Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4247–4255
Marseille, 11–16 May 2020

Cross-lingual Named Entity List Search via Transliteration

Aleksandr Khakhmovich[1,2], Svetlana Pavlova[1], Kira Kirillova[1], Nikolay Arefyev[1,2,3], Ekaterina Savilova[1]
[1]Samsung R&D Institute Russia
[2]Lomonosov Moscow State University
[3]National Research University Higher School of Economics
Moscow, Russian Federation
{a.hahmovich, s.pavlova, k.kirillova, n.arefyev, e.savilova}@partner.samsung.com

Abstract

Out-of-vocabulary words are still a challenge in cross-lingual Natural Language Processing tasks, for which transliteration from source to target language or script is one of the solutions. In this study, we collect a personal name dataset in 445 Wikidata languages (37 scripts), train Transformer-based multilingual transliteration models on 6 high- and 4 less-resourced languages, compare them with bilingual models from (Merhav and Ash, 2018) and determine that multilingual models perform better for less-resourced languages. We discover that intrinsic evaluation, i.e comparison to a single gold standard, might not be appropriate in the task of transliteration due to its high variability. For this reason, we propose using extrinsic evaluation of transliteration via the cross-lingual named entity list search task (e.g. personal name search in contacts list). Our code and datasets are publicly available online.
Keywords: Natural Language Processing, Transliteration, Named Entity Transliteration, Cross-Lingual, Multilingual, Transformer

1. Introduction

Transliteration, being a common way to adapt the out-of-vocabulary words, such as named entities of any types, from one language (script) to another, is an important component in many language processing tasks. A popular way to approach transliteration is to consider it a specific case of machine translation. Transliteration faces ambiguity problem due to large amount of language-specific phonetic variations. Another challenge, mostly specific for rare or less-resourced languages, lies in the field of data collection.

In this study[1], we apply a Transformer-based solution for multilingual transliteration as a core technology, and evaluate it in a cross-lingual named entity list search task. The contributions in this paper are summarized as follows:

- We collect and release a multilingual personal name dataset obtained from Wikidata, which covers 445 languages and 37 scripts. Scripts for basic pre-processing and train, dev, test split are supplied.
- We propose multilingual transliteration models which outperform bilingual models presented in (Merhav and Ash, 2018)[2] in English→target language transliteration experiment on multiple language pairs: English→Hebrew, English→Japanese, English→Katakana[3], English→Korean. We also test the models on 4 new less-resourced languages (Belarusian, Odia, Punjabi, Telugu) and obtain better results with multilingual models than with bilingual.
- We provide a linguistic analysis of the transliteration model predictions and reveal significant task-specific problems related to transliteration variability.
- We propose the cross-lingual named entity list search task, prepare use case-specific evaluation datasets, and use the devised transliteration model to solve it.

The rest of the paper is organized as follows: Section 2 gives the review of the previous work in the field of enabling multilingual translation/transliteration and using it as a search sub-task. Section 3 introduces our methods of collecting and processing data and gives a detailed description of the named entity transliteration datasets we are releasing with this paper. Section 4 describes our experimental setup, provides in-depth information on the conducted experiments in multilingual named entity transliteration, introduces the evaluation metrics, and combines the results of our experiments and transliteration error analysis. Section 5 presents named entity list search task definition, datasets, evaluation metrics and results. Finally, Section 6 presents our conclusion and determines the direction for future work.

2. Related Work

First, it is necessary to distinguish transliteration from other similar NLP tasks it can be easily confused with, which are transcription and translation.

Transcription is traditionally approached by researchers as a part of a speech-to-text problem, involving the transformation from the spoken language (sound) to its written form through a standardized sound representation (e.g. International Phonetic Alphabet). Thus, the goal of transcription is to capture phonetic or phonemic form of the word. For example, phonetic transcription of the English word *tree* is [tɹiː] using IPA.

Translation may be defined as a task of conveying the original meaning of a word in a given language by the lexical means of another language. Returning to the previous example, the translation for the English word *tree* into Russian is *дерево* [dʲerʲɪvə]. Note that phonetic representations of the English and Russian words are completely different.

[1]Current article repository: https://github.com/perspective-Alex/NE-transliteration-and-search.

[2](Merhav and Ash, 2018) article repository: https://github.com/steveash/NETransliteration-COLING2018/

[3]Detailed information on the choice to separate out personal names Katakana is provided in Section 3.2

Transliteration is the process of text conversion from one orthographic system, or script, to another (Rosca and Breuel, 2016), often, but not always, including the phonetic translation of the words in a source language to the equivalent words in the target language (Le and Sadat, 2018). The English *tree* (Latin script) could be transliterated into Russian as *три* [trʲi] (Cyrillic script). Transliteration could be considered a special case of translation, even though several transliteration systems can be adopted by a single language. For example, Revised Romanization, McCune-Reischauer and Yale Romanization are different transliteration systems representing Korean language in Latin script and English language specifically. For named entities, the distinction between translation and transliteration is especially thin because phonetic forms of proper names in source and target languages are similar, apart from the cases of historically established name equivalents (e.g. the Pope John Paul I's name in the Latin language is Ioannes Paulus I). We define our task as transliteration following the tradition of shared tasks within the Named Entity Workshops (Chen et al., 2018).

The earliest works in the field of automatic transliteration involved the pre-defined set of rules designed to convert the phonetic representation of proper nouns from one script to another (Knight and Graehl, 1997), following the example of rule-based machine translation. This approach, however, was inefficient in the way of handling language ambiguity. In the early work (Joshi et al., 2008) transliteration and translation were separated to achieve better performance in the task of cross-lingual location search; a combination of these was used to generate candidates for fuzzy search across the list of multilingual entities of "Location" type, covering the number of languages in different scripts. As shown in the mentioned paper, this approach helps to achieve maximum coverage despite transliteration ambiguity. (Jacquet et al., 2016) reports on a cross-lingual resource-based approach to the task of linking named entity acronyms and their extensions across languages. As only Roman-script languages were used, the number of possible common acronyms makes it possible to use string similarity distance for monolingual clustering. Translation probabilities obtained by a statistical machine translation model are used to handle entities having different written forms across languages via cross-lingual cluster aggregation.

With the rise of neural network-based methods in Natural Language Processing, a number of novel techniques have been proposed; for instance, a Neural Machine Translation (NMT) model proposed by (Johnson et al., 2016) introduces the idea of handling the translation between multiple languages by a single model; multilinguality is enabled by introducing a target language token at the beginning of the input sentence. Adding multilinguality boosts the model performance on less-resourced languages compared to the corresponding bilingual models. However, further experiments on multilingual NMT (Arivazhagan et al., 2019) show that for high-resourced languages multilingual models perform slightly worse than bilingual models. In (Roller et al., 2018) many-to-one translation enabled by sharing the encoder across multiple languages is an optional step in the task of cross-lingual biomedical term search.

Finally, (Merhav and Ash, 2018) introduce the newest Transformer method to the named entity transliteration task and prove its effectiveness by comparing its performance with an LSTM model and a traditional WFST model (Novak et al., 2012). Additionally, (Merhav and Ash, 2018) release a dataset consisting of named entities of person-type for 7 language pairs. We are building on the work done in (Merhav and Ash, 2018) by enabling multilinguality and expanding the target language pool.

3. Data

3.1. Wikidata

Wikidata entity labels in different languages were used to construct train, dev and test sets of personal names. Language labels were extracted from the latest at the time json Wikidata dump[4]. Only instances of humans were considered. The resulting dataset consisting of Wikidata IDs for human entities with corresponding names in different languages is available in the article repository. Below is a brief description of the dataset.

Altogether, there are more than 5 million entries of human entities with labels in 445 languages in our dataset. On average, entities have 11 labels in different languages (median = 7, std = 18.63). The minimal number of labels is 0 (for 433 entities), and the maximum is 413 (17 entities). Unfortunately, the large number of different language labels in most cases is achieved by copying English (or other prominent language) labels for all other languages, as is the case with, for example, *Claude Vaucher*, for whom all 413 labels are exactly the same.

If we count only labels which differ from one another at least by one symbol, the numbers are much less impressive: 1.87 different labels per entity (median = 1, std = 2.15). *Jesus Christ* has the largest number of different labels — 184. This presents a challenge in constructing datasets for the transliteration task: we cannot take a particular language label at face value and need to at least check whether the label is in the corresponding script. This step was performed for all experimental languages in this article.

On average, languages have labels for 130K people (median = 22,903, std = 501,017.6), however the variation is huge. Six languages are represented by only one label; there are more than 4 million labels in English, which is the largest language in Wikipedia. The list of the 30 largest languages by the number of labels for human entities is presented in Table 4 (Appendix A).

Transliteration is particularly interesting when the source and target languages are represented by different scripts. Transliteration within one script is still reasonable. For example, within the Latin script, *Jesus Christ* (English) is *Ježiš Kristus* in Slovak, and within Cyrillic, *Ісус Хрыстос* in Belarusian and *Иисус Христос* in Russian. However, the Slovak version of the Latin script is readable to English speakers, and Russian is readable to Belarusian. Therefore,

[4]https://dumps.wikimedia.org/wikidatawiki/entities/, October 2019.

efforts in transliteration have mostly been focused on converting one script into another.

We used Unicode character ranges[5] to check which scripts were used in the collected Wikidata entity labels. Altogether, there are more than 800 combinations of different scripts and character types — apart from writing system symbols themselves, also punctuation, numbers, diacritics and all kinds of extensions and supplements. The number of entity labels for 30 largest scripts is presented in Table 5 in Appendix A. Combinations of scripts (e.g. Latin, CJK[6] and Hiragana in *Monday* 満ちる) were not counted. Scripts with less than 100 entity labels include Unified Canadian Aboriginal Syllabics (82), Cherokee (42), NKo (9), Tifinagh (4), Meetei Mayek (3), Mongolian (1) and Bopomofo (1).

The script counts illustrate the problem of using Wikidata labels for transliteration to and from less-resourced languages. For example, although there are nominally more than 20K entity labels for Cherokee language, only 42 of them are in Cherokee script — the rest are in Latin. At the same time, if a model were devised for transliteration from Latin script (English) to Cherokee, the data could be enriched greatly. In this article, we focus on less sparse languages and scripts, and leave this challenge for future researchers.

3.2. Transliteration Dataset

For the transliteration task, we took existing datasets of paired (source language→target language) single-word named entities from (Merhav and Ash, 2018) for designated high-resource languages — Arabic (ar), Chinese (zh), Hebrew (he), Japanese (ja), Katakana (kat), Korean (ko), Russian (ru). Names written in Katakana syllabary were analyzed separately, as well as inside the Japanese dataset, due to expected significant difference between transliteration quality for logographic characters (e. g. Chinese hanzi), syllabic (Japanese kana), and alphabetic (Latin letters) characters. From now on we refer to Katakana as 'language' for the sake of uniformity and brevity. We chose not to augment the (Merhav and Ash, 2018) datasets with newly acquired data for the purpose of direct comparison. To evaluate our models on less-resourced languages, we extracted paired (English→target language) personal names from the Wikidata dataset in four languages with relatively few human entity labels: Belarusian (be), Odia (or), Punjabi (pa), Telugu (te). Basic pre-processing was applied to names in all experimental languages:

- As a method of alignment, only pairs of personal name labels with the same number of words in English and the target languages were used.
- If a label contained a comma, everything before the comma was moved to the end of the label, and the comma itself was deleted (relevant for *the surname, first name* label format).
- Various punctuation marks were deleted from the paired labels.
- Korean (Hangul) syllabic blocks were split into jamos with `jamotools`.

The splits between train (64%), dev (16%) and test (20%) datasets are presented in Table 1.

4. Neural Approach to Transliteration

Transliteration task, as mentioned above, could be considered a sequence-to-sequence transformation task. Thus we apply the Transformer architecture (Vaswani et al., 2017) following (Merhav and Ash, 2018). Transformer was implemented using Tensorflow Keras API.

In this architecture, the model is represented with a set of encoder and decoder layers. Encoder layers get a sequence of characters (letters for alphabetic or syllables for syllabic writing systems) of a named entity and transform it into internal representation which is fed into a decoder. Final transliteration is produced from the last decoder layer one character at a time considering previous timesteps. As usual in such tasks, the model minimizes negative log likelihood loss.

The model has a certain number of hyperparameters. Firstly, there are architecture-oriented ones: number of layers in encoder and decoder, number of heads used in each layer's multi-head attention component, and hidden size. Secondly, there are training-oriented hyperparameters, such as learning rate schedule and level of regularization through dropout and "l2" weights regularization. Dropout is used in three different places excluding repeated encoder or decoder layers: after the embedding layers (`embedding dropout`), after the attention layers (`attention dropout`), and inside fully connected layers (`relu dropout`). We use double-size hidden dimension inside the fully connected layer.

Our experiments were not purposefully focused on hyperparameter optimization. We varied 2 levels of regularization, 3 levels of learning rate value (with 5 times difference) used with Adam optimizer (Kingma and Ba, 2014) (default for Tensorflow implementation parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e-08$) with the following update strategy: 1 epoch of linear warmup and further constant value. We also tried different number of layers and attention heads. Setup for our best models is described in Section 4.3

4.1. Main Experimental Setup

Transliteration task implies transformation between two languages, but it is possible to allow transformation from one to multiple languages at the same time. Accordingly, we use English as the source language and the high- and less-resourced languages described in Section 3.2 as target languages. Let us denote this setup as the *English2All* experiment. Enabling multilinguality requires the following changes in the model.

Firstly, instead of traditional <start> token, we pass source and target language tokens to the encoder and decoder respectively. They are equally represented as <language>. The source language token could help the encoder to distinguish languages/syllabaries that potentially share a script

[5] https://www.unicode.org/Public/UCD/latest/ucd/NamesList.txt.
[6] Chinese, Japanese, Korean and Vietnamese characters.

Target language	Ar	He	Ja	Kat	Ko	Ru	Zh	Be	Or	Pa	Te
Train	42K	32K	64K	63K	31K	105K	39K	13K	4.7K	5.8K	6.8K
Dev	10K	8K	16K	16K	8K	26K	10K	3.2K	1.2K	1.4K	1.7K
Test	13K	10K	20K	20K	10K	33K	12K	4.1K	1.5K	1.8K	2.2K

Table 1: Size of the train, dev and test datasets for each language.

(in our case, Japanese language and Katakana syllabary, or Belarusian and Russian languages). The target language token helps the decoder to determine which part of the mixed target vocabulary has to be used during per-token generation of the target sequence. It serves as the only source of this information.

Secondly, we have to combine our train bilingual datasets into single train set. We use different techniques:

- **default**: bilingual train sets remain unchanged;
- **addition of sample duplicates**: we identify the length of the largest bilingual train dataset (English→Russian) and add duplicates of samples in other bilingual datasets up to this bound;
- **slicing**: we identify the length of the smallest bilingual train dataset (English→Odia) and slice other bilingual train sets down to this bound.

After applying one of these techniques, we combine datasets together and shuffle before training. Finally, we increase the default batch size (25) by the number of times equal to the number of used target languages.

In order to compare the train data combination effect on model performance, we trained models with each of these techniques. Moreover, assuming the potential sensitivity of the learning rate to the different amount of observed data in each epoch, we tried three variations of learning rate value for each of the techniques. For these 9 training procedures, we set the approximately equal number of weights update steps per epoch using different batch sizes. Results are shown in Section 4.3.

It is worth noting that our strategy allows to set up the reverse experiment: *All2English*. As we have not sufficiently considered model performance for the reverse transliteration directions, we do not provide any results. However, this is the immediate focus for our future work.

4.2. Evaluation Metrics

The main transliteration metric we use to compare the models on test sets is *explicit accuracy*, computed as the percentage of samples where prediction exactly equals the corresponding ground truth label. In other words, *explicit accuracy* (EA) = $1 - WER$, where WER stands for *word error rate* metric used in (Merhav and Ash, 2018). The k-best metrics are defined as the percentage of samples where ground truth could be found in top k predictions, which are obtained using the beam search algorithm.

Explicit accuracy might be a too strict metric to measure actual performance in case of sequence generation tasks — particularly, in transliteration, which allows multiple equally correct variations. This issue is further discussed in Section 4.4. For this reason, we inspected another accuracy metric on the character (or token, from the model point of view) level. For each sample, let us denote token accuracy as the percentage of correctly matched corresponding tokens in prediction and ground truth. For k-best case, the maximum value across k predictions is taken. Averaging the results across all samples we get the final value. Token accuracy metric proved to be more robust and stable on the adjacent epochs than explicit accuracy.

4.3. Results

Train data combination methods comparison. As mentioned in Section 4.1, we tried different train data combination techniques during the *English2All* experiment. As expected, **slicing** method gives poor results in our case because the size of the smallest train set (Odia) is 20 times smaller than of the largest (Russian). The model is faced with the lack of training data.

We identified two best models (across overall 9) that are comparable in quality on the dev sets for each language pair. They have the same hyperparameters, including learning rate equal to 1e-4. The models differ only in the train data combination method: (**default** and **addition of duplicates**). Comparison is summarized in Figure 1.

The model which used the **addition of duplicates** method performed the same or better on 7 high- and less-resources languages and worse on 4 high-resourced languages: Russian, Katakana, Japanese and Korean. The possible reason might be the train size difference between the languages. As mentioned before, the maximum size is 20 times higher than the minimum. Consequently, during 1 epoch for 1 sample in, for example, Russian as the target language, the model sees 20 samples in Odia. The model is trained harder on less-resourced languages than on high-resourced. This reason probably explains another fact: the **addition of duplicates** model performed better on 4 less-resourced languages (right-most in Figure 1), because the **default** model saw samples from these languages quite rarely during 1 epoch.

For further usage for our best model, we chose the **addition of duplicates** method aiming to get better results on less-resourced languages.

Bilingual versus Multilingual model performance. Top-1, 2 and 3 explicit accuracies (EA) on test sets for each target language are provided in Table 2. We compare results of bilingual models from (Merhav and Ash, 2018) (reproduced using the article's code) and our multilingual models with **addition of duplicates** train data combina-

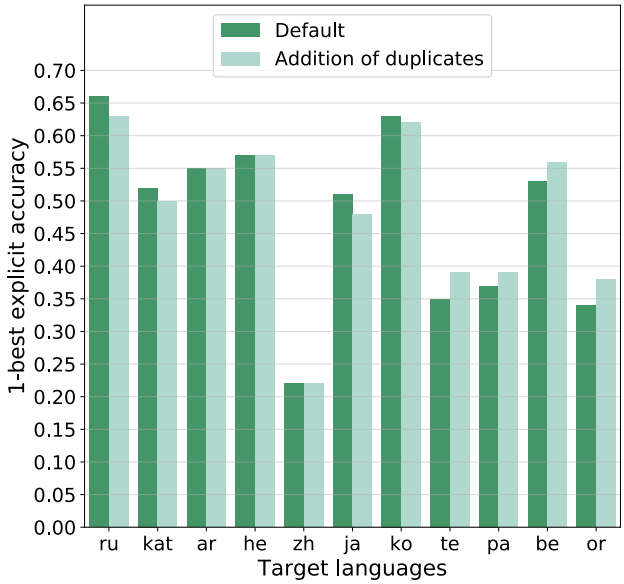

Figure 1: Comparison of multilingual models performance using **default** and **addition of duplicates** train data combination methods.

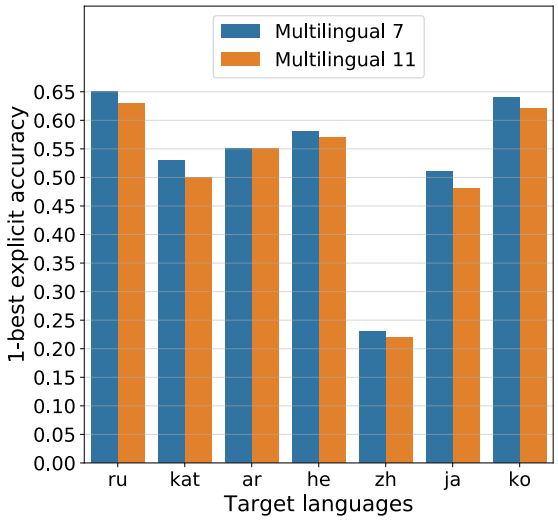

Figure 2: Comparison of *Multilingual-7* and *Multilingual-11* models performance on 7 high-resourced languages.

tion method in 2 versions: with 7 high-resourced target languages *(Multilingual-7)* and its expanded version with 4 less-resourced target languages *(Multilingual-11)*. Multilingual models were both trained during 200 epochs with the following hyperparameters: dropout (all - embedding, attention and relu) 0.2, "l2" lambda coefficient equal to 1e-5, 2 heads inside each attention layer, 2 layers for both decoder and encoder, 128 hidden size, learning rate during constant stage of update strategy equal to 1e-4.

The best result for each language pair is bolded (apart from Arabic, for which the model performance did not differ significantly). The multilingual model with 7 incorporated target languages achieves equal to the bilingual (Merhav and Ash, 2018) results for Russian, Arabic and Chinese, and outperforms the bilingual model for Hebrew, Katakana, Japanese and Korean. The *Multilingual-11* model surpasses the bilingual on all less-resourced languages.

Multilingual models comparison. Figure 2 compares the *Multilingual-7* and *Multilingual-11* model performance for 7 high-resourced languages. For each target language apart from Arabic, top-1 explicit accuracy is lower for *Multilingual-11* than *Multilingual-7*. For Arabic, the accuracies are approximately the same.

The multilingual models were trained with the same hyperparameters and number of training epochs. The same tendency could be observed in loss dynamics graphs on dev sets during training for each target language: *Multilingual 11* trains slower even though we provided equal number of gradient updates per epoch for each training procedure. Evidently, the addition of 4 new languages requires changes to the training procedure (more training epochs, more optimal learning rate schedule) and most likely increasing the model capacity. For instance, it is possible that decoder starts to suffer from the lack of trainable parameters while trying to provide equally decent result for all languages. Increasing the hidden size, the number of decoder layers and attention heads could be useful even without careful search of more optimal hyperparameters.

4.4. Error analysis

To analyse the model errors, we have sampled 200 random name pairs (English-Korean and English-Russian) for which the most probable three predictions of the model did not contain ground truth, i.e. the erroneous predictions according to the top-3 accuracy metric. We looked at 10 predictions for each of the pairs. Below we describe the most frequent reasons underlying the errors.

The biggest challenge in the transliteration task is that letters (or sounds) of one script (or language) could be transformed into another in many different ways — there is no one-to-one correspondence between languages (and scripts). For example, *Maples* in English corresponds to *Мэйплс* in Russian for *Holly Maples*, to *Мэйплз* for *Marla Maples*, and to *Мейплс* for *Dillon Maples*. All of the transliteration variants are valid; the only difference is that historically each option was attached to a particular person. Moreover, these are not all possible options for the transliteration of the name *Maples* — there are about 20 of them if we only try to approximate English pronunciation, and many more if we suppose that Maples is actually a non-English name. However, in our dataset only few of the variations are present. This fact explains the majority of errors of the model.

Another challenge is that sometimes a name in one language is barely recognisable in another — in this case, it is more accurate to talk about translation. For example, *James* in

Source and target languages	Model	1-best EA	2-best EA	3-best EA
English→Arabic	Bilingual	0.55	0.7	0.76
	Multilingual 7	0.55	0.69	0.76
	Multilingual 11	0.55	0.69	0.75
English→Chinese	Bilingual	0.22	0.33	0.39
	Multilingual 7	**0.23**	**0.33**	**0.39**
	Multilingual 11	0.22	0.33	0.38
English→Hebrew	Bilingual	0.56	0.72	0.78
	Multilingual 7	**0.58**	**0.74**	**0.79**
	Multilingual 11	0.57	0.72	0.78
English→Japanese	Bilingual	0.48	0.64	0.7
	Multilingual 7	**0.51**	**0.67**	**0.74**
	Multilingual 11	0.48	0.64	0.71
English→Katakana	Bilingual	0.49	0.64	0.71
	Multilingual 7	**0.53**	**0.68**	**0.74**
	Multilingual 11	0.5	0.65	0.72
English→Korean	Bilingual	0.62	0.75	0.81
	Multilingual 7	**0.64**	**0.77**	**0.82**
	Multilingual 11	0.62	0.75	0.8
English→Russian	Bilingual	**0.65**	**0.78**	**0.83**
	Multilingual 7	0.65	0.77	0.82
	Multilingual 11	0.63	0.76	0.81
English→Belarusian	Bilingual	0.52	0.66	0.72
	Multilingual 11	**0.55**	**0.71**	**0.76**
English→Odia	Bilingual	0.37	0.5	0.58
	Multilingual 11	**0.38**	**0.53**	**0.6**
English→Punjabi	Bilingual	0.35	0.49	0.56
	Multilingual 11	**0.39**	**0.53**	**0.61**
English→Telugu	Bilingual	0.37	0.5	0.57
	Multilingual 11	**0.39**	**0.54**	**0.61**

Table 2: Bilingual and multilingual model comparison according to the *k*-best exact accuracy (EA) metric in *English2All* experiment.

English corresponds to *Джакомо* [d zɐkomə] in Russian for the name *James Salomoni* due to the person's Italian origin. Other times, the name's phonetic form is reproduced in the target language: *Foucault* in English corresponds to *Фуко* [fʊko] in Russian and 푸코 [puko] in Korean. This is a highly language-specific process. The name's exact phonetic form cannot be determined by its graphical form alone, without the information about the person's (or the name's) origin.

Sometimes, not the whole name is "translated" into another language (script), but a part of it which carries separate meaning (i.e. morpheme). For example, Kazakh suffix for the masculine patronymic *-uly* corresponds to Russian *-вич*: *Jurynuly* [zurınulı] — *Журинович* [zurinovitʃ]. The synonymous morphemes rarely have similar phonetic form, unless the languages in question are related. That is why the model would most probably be mistaken in such cases. However, sometimes the model is able to handle them. For example, for the Polish surname *Bohuszewiczówna* [bɔxuʃɛvitʃuvna] (which is a feminine form of the surname *Bohuszewicz* [bɔxuʃɛvitʃ] for an unmarried woman), the ground truth Russian label is *Богушевич* [boguʃɛvitʃ]. Polish surnames are evidently translated into Russian without the feminine suffix. The first prediction of the model is the direct transliteration *Богушевичовна* [boguʃɛvitʃovna]; the seventh, however, is *Богушевич* — the correct one.

Around one tenth of the inspected errors in the English-Russian pairs constitute wrongly aligned names where, for example, in the English label the first name follows the surname, and in the Russian — the surname follows the first name (*Syuji Takahara — Такахара Сюдзи*). Most of these cases pertain to the names of Japanese origin. There were almost no such cases for English-Korean pairs.

5. Cross-lingual named entity list search

Task. We propose the cross-lingual named entity list search task as another application for the transliteration technology. The task comprises searching for an entity given in a source language (script) in a list of target language (script) named entities. The real-world use case for such a task would be personal name search in phone or social network contacts or geographical name search in geographical databases.

Data. For this task, a pool of paired (English→target language) personal names was formed from the newly acquired Wikidata human entity labels for each of the ten languages discussed above (Japanese and Katakana were distinguished as before). The names any part of which was used in the train

Target language	Exact match accuracy			Fuzzy match accuracy		
	Bilingual mean, std	Multilingual-11 mean, std	p-value	Bilingual mean, std	Multilingual-11 mean, std	p-value
Arabic	0.25, 0.04	0.26, 0.04	0.022	0.88, 0.03	0.88, 0.04	0.099
Chinese	0.06, 0.02	0.06, 0.02	0.439	0.42, 0.05	0.41, 0.05	0.064
Hebrew	0.30, 0.05	0.31, 0.04	0.052	0.83, 0.04	0.83, 0.04	0.641
Japanese	0.20, 0.04	0.21, 0.04	0.057	**0.56, 0.05**	**0.58, 0.05**	**<0.001**
Katakana	0.26, 0.04	0.27, 0.04	0.119	0.75, 0.04	0.75, 0.04	0.873
Korean	0.31, 0.05	0.31, 0.04	0.289	**0.66, 0.05**	**0.67, 0.04**	**<0.001**
Russian	0.42, 0.05	0.42, 0.05	0.908	0.86, 0.03	0.86, 0.04	0.075
Belarusian	0.36, 0.04	0.37, 0.05	0.002	**0.78, 0.03**	**0.83, 0.04**	**<0.001**
Odia	**0.25, 0.03**	**0.27, 0.03**	**<0.001**	**0.74, 0.04**	**0.76, 0.03**	**<0.001**
Punjabi	**0.24, 0.04**	**0.27, 0.04**	**<0.001**	**0.75, 0.04**	**0.78, 0.03**	**<0.001**
Telugu	0.28, 0.04	0.28, 0.04	0.897	**0.75, 0.04**	**0.77, 0.04**	**<0.001**

Table 3: Bilingual (Merhav and Ash, 2018) and *Multilingual-11* models accuracy comparison on contacts datasets for different target languages.

or development samples of the transliteration task were filtered out. Personal names could be single- or multi-word: the name length varied from 1 to 6, with mean 1.82 words and median 2. The resulting size of the name pool for each target language is presented in Table 6 in Appendix A). 100 lists of 100 names were randomly sampled from each pool without replacement. All lists of names are available in the article repository. These name lists model the personal contacts use case of the task, thereby we define them as contact lists.

Procedure. A probability-ordered list of ten transliteration candidates from English to target language was obtained separately for each word of the names in each of the lists using two transliteration models: bilingual from (Merhav and Ash, 2018) and the 11-language multilingual model discussed in this article.

While the chosen transliteration models were both trained on single-word names, named entity list search task implies mostly multi-word name search. One of the possible ways to obtain a ranked list of multi-word predictions is to multiply prediction probabilities of each word to simulate multi-word prediction probability. However, the bilingual model from (Merhav and Ash, 2018) returns predictions ranked by probability in descending order, but doesn't return actual probability values. That is why, for the sake of model comparison, we use a simpler method to get top k multi-word predictions: single-word predictions of one rank were combined to create a multi-word prediction with the same rank.

After getting multi-word transliteration candidates, we searched for these candidates in the target language list using two modes: exact matching and fuzzy matching. For exact match, the transliteration candidate and the ground truth name were matched if the candidate was found in the list and matched the true name exactly. For fuzzy match, the transliteration candidate and the ground truth name were matched if the candidate could be obtained from ground truth by performing no more than one edit operation (delete, add or substitute character) upon each constituting word. For example, the candidate *Юри Долгарукий* was matched with the ground truth name *Юрий Долгорукий* (English source: *Yuri Dolgorukiy*). Accuracy is defined as the percentage of correctly matched names in the list.

Results. Mean and standard deviation of the exact match and fuzzy match accuracies across 100 lists for each target language are presented in Table 3. The differences between match accuracies for bilingual and multilingual models were checked for normality with D'Agostino and Pearson's normality test with α=0.001 (`normaltest` from the `scipy` package). All difference distributions were found to be normal. The significance of the differences between match accuracies for bilingual and multilingual models was tested with Student's paired t-test with α=0.001 (`ttest_rel` from the `scipy` package).

Several tendencies could be observed:

- Performing fuzzy match search greatly increases the chances of finding the name. The biggest accuracy increase of 63 pp was reached for Arabic.
- Bilingual and multilingual models demonstrate more differences for less-resourced languages than for high-resourced languages.
- The match accuracy was higher for the multilingual model compared to bilingual when the target languages were Korean, Japanese, Belarusian and Telugu (fuzzy match), Odia and Punjabi (exact and fuzzy match).
- The comparatively large increase in match accuracy for Belarusian could potentially be due to transfer from Russian, as the languages share Cyrillic script.

The above-mentioned differences are statistically significant. However, relatively large standard deviations of the metrics should be taken into account.

Besides correctly found names, there could potentially be incorrectly matched ones in our lists for very similar names. For example, *Mari* and *Marie* could have the same Russian transliteration *Мари*, and therefore could be incorrectly

matched. However, the number of these cases was extremely small: usually 0, maximum was reached for Chinese with the mean 3% incorrectly matched names. Most of the names contained two or more words and the probability of incorrect match falls with each additional word in the name.

6. Conclusion

According to our contribution points, we provide a large multilingual and multiscript person-type named entity dataset, which could be used for various purposes, including named entity transliteration. Using multilingual transliteration models, we improve results on test datasets for high-resourced languages from (Merhav and Ash, 2018) and introduce benchmarks for 4 new less-resourced languages. We provide analysis of transliteration errors which explains some considerable task-specific problems. We propose a new point of view on the cross-lingual named entity search task based on transliteration. In this task, the multilingual model demonstrates performance comparable to the bilingual model on most high-resourced languages, and slightly outperforms the bilingual model on less-resourced languages. While some improvement could be reached with the multilingual approach, it does not, evidently, solve the problem of small training sets.

An immediate future focus for our work could be implementation of language-specific regularization inside multilingual transliteration models. This might prevent the model from overfitting on some language pairs and underfitting on others.

7. Acknowledgements

We thank the three anonymous reviewers of this paper, as well as our colleagues at Samsung R&D Institute Russia for valuable comments and suggestions. The contribution of Nikolay Arefyev to the paper was partially done within the framework of the HSE University Basic Research Program funded by the Russian Academic Excellence Project '5-100'.

8. Bibliographical References

Arivazhagan, N., Bapna, A., Firat, O., Lepikhin, D., Johnson, M., Krikun, M., Chen, M. X., Cao, Y., Foster, G., Cherry, C., Macherey, W., Chen, Z., and Wu, Y. (2019). Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Chen, N., Duan, X., Zhang, M., Banchs, R. E., and Li, H. (2018). News 2018 whitepaper. In *Proceedings of the Seventh Named Entities Workshop*, pages 47–54, Melbourne, Australia, July. Association for Computational Linguistics.

Jacquet, G., Ehrmann, M., Steinberger, R., and Väyrynen, J. (2016). Cross-lingual linking of multi-word entities and their corresponding acronyms. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 528–535, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Joshi, T., Joy, J., Kellner, T., Khurana, U., Kumaran, A., and Sengar, V. (2008). Crosslingual location search. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 211–218, New York, NY, USA. ACM.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Knight, K. and Graehl, J. (1997). Machine transliteration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98/EACL '98, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.

Le, N. T. and Sadat, F. (2018). Low-resource machine transliteration using recurrent neural networks of Asian languages. In *Proceedings of the Seventh Named Entities Workshop*, pages 95–100, Melbourne, Australia, July. Association for Computational Linguistics.

Merhav, Y. and Ash, S. (2018). Design challenges in named entity transliteration. *CoRR*, abs/1808.02563.

Novak, J. R., Minematsu, N., and Hirose, K. (2012). WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49, Donostia–San Sebastián, July. Association for Computational Linguistics.

Roller, R., Kittner, M., Weissenborn, D., and Leser, U. (2018). Cross-lingual candidate search for biomedical concept normalization. *CoRR*, abs/1805.01646.

Rosca, M. and Breuel, T. (2016). Sequence-to-sequence neural network models for transliteration. *CoRR*, abs/1610.09565.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Appendices

A. Data tables

#	Language	Count	#	Language	Count	#	Language	Count
1	English	4,754,826	11	Swedish	1,074,287	21	Traditional Chinese	542,060
2	Asturian	4,594,059	12	Danish	978,685	22	Finnish	531,716
3	Spanish	4,378,313	13	Bokmål	960,716	23	Arabic	497,619
4	Dutch	4,273,706	14	Nynorsk	861,414	24	Japanese	491,018
5	French	2,936,057	15	Irish	804,269	25	Classical Chinese	422,375
6	Slovene	2,394,331	16	Russian	798,517	26	Welsh	406,156
7	German	2,333,118	17	Portuguese	789,861	27	Galician	335,131
8	Catalan	1,951,776	18	Chinese	750,801	28	Czech	310,812
9	Italian	1,456,619	19	Hungarian	685,701	29	Brazilian Portuguese	277,547
10	Albanian	1,257,632	20	Polish	566,614	30	Romanian	274,172

Table 4: The largest languages by the number of labels for human entities.

#	Script	Count	#	Script	Count	#	Script	Count
1	Latin	52,546,892	11	Bengali	54,146	21	Gujarati	2,456
2	CJK	2,151,368	12	Thai	31,535	22	Hiragana	2,212
3	Cyrillic	1,280,738	13	Georgian	24,208	23	Sinhala	1,825
4	Arabic	813,467	14	Tamil	23,390	24	Ethiopic	1,785
5	Katakana	238,443	15	Malayalam	16,484	25	Tibetan	795
6	Hebrew	126,187	16	Telugu	14,745	26	Khmer	594
7	Hangul	120,036	17	Gurmukhi	8,818	27	Lao	502
8	Greek and Coptic	76,604	18	Oriya	8,589	28	Ol Chiki	460
9	Armenian	63,342	19	Kannada	8,434	29	Thaana	231
10	Devanagari	56,442	20	Myanmar	2,779	30	Syriac	134

Table 5: The largest scripts by the number of labels for human entities.

Target language	Ar	Be	He	Ja	Kat	Ko	Or	Pa	Ru	Te	Zh
Name pool size, words	12,153	684	3,691	13,513	10,334	5,865	308	419	12,656	623	28,470

Table 6: The number of personal names in the sample pool per language for the named entity list search task.