A Generalized Method for Automated Multilingual Loanword Detection

Abhijnan Nath[*], **Sina Mahdipour Saravani**[*†], **Ibrahim Khebour**,
Sheikh Mannan, **Zihui Li** and **Nikhil Krishnaswamy**
Situated Grounding and Natural Language (SIGNAL) Lab
Department of Computer Science, Colorado State University
Fort Collins, CO, USA
abhijnan.nath@colostate.edu, sina@cs.utah.edu[†],
{ibrahim.khebour,sheikh.mannan,nkrishna}@colostate.edu

Abstract

Loanwords are words incorporated from one language into another without translation. Suppose two words from distantly-related or unrelated languages sound similar and have a similar meaning. In that case, this is evidence of likely borrowing. This paper presents a method to automatically detect loanwords across various language pairs, accounting for differences in script, pronunciation and phonetic transformation by the borrowing language. We incorporate edit distance, semantic similarity measures, and phonetic alignment. We evaluate on 12 language pairs and achieve performance comparable to or exceeding state of the art methods on single-pair loanword detection tasks. We also demonstrate that multilingual models perform the same or often better than models trained on single language pairs and can potentially generalize to unseen language pairs with sufficient data, and that our method can exceed human performance on loanword detection.

1 Introduction

Throughout history, words and phrases have been exchanged between languages around the world (Weinreich, 1954). This can obscure genetic relations between languages (e.g., many people erroneously believe English and French are more closely related than they are) but may also increase comprehension of foreign languages by monoglots (e.g., written French is often partially comprehensible by English speakers).

As Zhang et al. (2021) observe, detecting that a word is loanword is conceptually straightforward: both similar sound and meaning suggests too great a coincidence for different words to have converged by chance[1]. Detecting loanwords computationally has therefore relied on pairwise similarity measures based on transliteration detection and edit distance. However, foundational work in linguistic borrowing, e.g., by Haugen (1950) and Betz (1959), established that when borrowing words into a recipient language, speakers of that language will reproduce existing linguistic patterns when using new words, and the patterns that recipient speakers impose upon a borrowed word vary across time (Köllner and Dellert, 2016), and language pairs. Some languages may adopt a word without much phonetic change due to already-similar phonotactics. Others may fit imported words into a rigid sound pattern, with sometimes significant transformation. Still others may change the meaning. Changes are particular to the language pair, so automatically detecting loanwords between *arbitrary* languages is challenging. However, if successful, such capabilities would also provide benefits to many other NLP tasks such as machine translation, coreference, and named-entity recognition (NER), because common vocabulary, coreferents, or named entities across languages may often be loanwords.

Here, we present a novel method for automated loanword detection between arbitrary language pairs. We build upon existing edit distance-based approaches, incorporate semantic similarity metrics from multilingual language models MBERT (Devlin et al., 2019) and XLM (Conneau et al., 2020), and a method of assessing alignment of phonemes between donor words and loans to account for differences in phonotactics between the relevant languages. We also present and evaluate on the WikLoW (Wiktionary LoanWord) Dataset, currently consisting of 13 language pairs with a high density of loanwords and 3 further language pairs with a lower density of loanwords. We also provide a methodology for expanding the dataset to new language pairs. We demonstrate that our method to detect loanwords across all language pairs in the dataset performs comparably to or better than

[*] These authors contributed equally to this work.

[†] This work performed at Colorado State University.

[1] There are exceptions, e.g., Persian *bad* vs. English "bad" and Mbabaram *dog/dúg* vs. English "dog". These are beyond the scope of this paper.

Proceedings of the 29th International Conference on Computational Linguistics, pages 4996–5013
October 12–17, 2022.

existing methods on language-specific loanword detection tasks, that multilingual models can perform better than models trained on individual language pairs, even on data from that pair itself, and that our model can also exceed human performance.[2]

Our method supports both loanword detection and construction of parallel corpora of loanwords for other tasks. Our conclusions suggest that there are some general principles of loanword detection that can be picked up by machine learning models independent of specific languages, and we propose follow-up challenges for NLP research in this area.

2 Related Work

Prior approaches to detecting loanwords computationally follow the intuition mentioned above: that if two words in otherwise not closely related languages have similar meaning and sound similar, then this is likely evidence of borrowing. Van Der Ark et al. (2007) use a Levenshtein-distance based approach to identify language groups and loanwords among languages of Central Asia.

Delz (2013)/Köllner (2021) proposes theoretical approaches to loanword identification based on phylogenetic methods. Zhang et al. (2021) also point out an issue we address herein: loanwords may be transformed to fit the borrowing language's phonology and phonotactics, so pronunciation similarity may be a weaker than ideal method.

Existing data resources relevant to loanwords include the the Automated Similarity Judgment Project (ASJP) database (Brown et al., 2008) and the World Loanword Database (WOLD) (Haspelmath and Tadmor, 2009). Our data source is Wiktionary, which has previously been used in related etymological tasks by De Melo (2014) and Sagot (2017).

One thing we should note is that much work in computational loanword detection and similar tasks is targeted at a specific language or group of languages, e.g., Romance (Cristea et al., 2021; Tsvetkov and Dyer, 2015), Japanese (Takamura et al., 2017), Uyghur (Mi et al., 2014, 2018, 2020, 2021), Spanish (Álvarez-Mellado and Lignos, 2022), Central Asian languages (Van Der Ark et al., 2007), or Turkic and Indo-Iranian (Zhang et al., 2021). Our approach attempts to address the problem at a multilingual level. We use and extend existing work in phonological processing by the NLP community, including the Epitran (Mortensen et al., 2018) and PanPhon (Mortensen et al., 2016) packages for representing phonetic and articulatory features. We incorporate semantic similarity measures from multilingual language models MBERT and XLM, and develop a method of scoring the level of alignment of phonemes between a donor and a loanword to account for differences in language-specific phonology and phonotactics. Our approach in principle supports loanword detection on any pair of languages supported by the upstream packages/models Epitran, MBERT, and XLM, but we discuss how we have (Sec. 3) and can (Sec. 8) also extend our approach to languages that are not at present covered by all of these.

A work at a similar scale, albeit on the slightly different task of *cognate classification*, is Jäger (2018), which evaluates PMI and SVM-based methods over the ASJP database. Cognate detection work generally uses similar methods to those we use here, e.g., semantic and phonetic similarity (Kondrak, 2001), orthographic distance (Mulloni and Pekar, 2006) combined with semantic information (Labat and Lefever, 2019; Lefever et al., 2020), or global constraints (Bloodgood and Strauss, 2017). Work in translation lexicons (e.g., Schafer and Yarowsky (2002)) is also relevant, for the hybrid approach to similarity metrics.

Loanword detection may be useful for phylogenetic reconstruction, like cognate detection (Rama and List, 2019). However, cognates are valid for reconstructing common ancestry; loanwords are not. For historical reconstruction, the two must be separated. Many in the NLP community adopt a definition of "cognate" that subsumes loanwords (e.g., Kondrak (2001)). We do not adopt this definition, and use the linguistic definition that treats loanwords and cognates as distinct.

3 Data Collection

The WikLoW dataset is collected using the process outlined in this section, which can be run for any pair of languages that have loans between them catalogued in Wiktionary, making it easy to expand to new data. We begin by collecting data from Wiktionary categories of the form `[Recipient]_terms_borrowed_from_[Donor]`[3]. Each link in the category is scraped for a loanword in the recipient language and the original form of that word in the donor language.

[2] The codebase is provided at https://github.com/csu-signal/loan-word-detection.

[3] e.g., https://en.wiktionary.org/wiki/Category:Polish_terms_borrowed_from_French

Table 1 shows the language pairs currently contained in the WikLoW dataset, and the number of loans for that pair. There is no global definition of a "low-resourced" language, as this is task-dependent, but we have intentionally tried to represent languages that are not well-represented in large corpora like CC-100 (Conneau et al., 2020). We hereafter refer to language pairs using the format "borrower-donor," e.g., "Hindi-Persian" to refer to Hindi words borrowed from Persian. The directionality between the two languages is important to the pair definition, as only words loaned from the donor language to the borrower are properly considered loanwords. If the direction of the languages were flipped, not only would the class labels be different (the donor word loaned into borrower would not be considered a loanword in the donor language), but while the phonetic and semantic similarities (Secs. 4.2 and 4.3) would probably be the same, the alignment score (Sec. 4.4) would not be, since the output label when training that network is the loanword status, which would be likewise flipped.

Borrower	**Donor**	**# loans**
English	French	5074
English	German	2942
Indonesian	Dutch	2665
Polish	French	2055
Romanian	French	2000†
Kazakh*	Russian	1809
Persian	Arabic	1526
Romanian	Hungarian	1460
German	French	1365
Hindi*	Persian	1249
Finnish	Swedish*	1242
Azerbaijani*	Arabic	1116
Mandarin	English	960
Hungarian	German	532
German	Italian	249
Catalan*	Arabic	94

Table 1: Loanword counts per language pair.

*Languages with < 2 billion tokens in the CC-100 corpus.
†Subset of total available loans used.

We also scrape the Wikipedia page listing languages by writing system[4], to include the script name for each language in our datasets. This allows us to filter out words not written in the typical script of the recipient language. For example, some Chinese "loanwords" from English are incorporated keeping the Latin script intact; we don't need machine learning to tell us that these are borrowed terms. Having script information also proves beneficial in later experiments (see Sec. 5).

We also collect all the available lemmas in the donor language, which we use later to calculate the closest phonetic neighbors for each loanword. We also collect homonyms for each loanword where available; homonyms are considered those words that have more than one etymology, where one is a loan from the relevant donor language[5].

Using the Epitran package (Mortensen et al., 2018), we transliterate both loans and original words into the International Phonetic Alphabet (IPA). The Epitran package can be extended to support new languages, as we did here in the case of Finnish, using Omniglot[6] as a resource. Epitran is not a perfect mapping to real pronunciation, especially in the case of abjads such as Arabic script, a point of relevance later (Sec. 4.4, Sec. 7.1).

Having gathered positive examples of loanwords, we need to gather sufficient negative examples to both train an algorithm, and to try and fool the trained algorithm. Negative examples can be:

- **Synonyms**: words with similar meaning to a loanword but pronounced differently, e.g., "driver" vs. *chauffeur*.
- **Hard negatives**: closest phonetic neighbors to a loanword that have different meaning, e.g., "annex" vs. *ânesse*.
- **Randoms**: random pairings where the two words have no discernible phonetic or semantic relationship.

To create the **synonyms** dataset, we take a list of 440 English words, each of which has multiple synonyms associated with it. With the Google Translate API, we translate the main word into one language from our current relevant pair, and each synonym into the other. We then construct word pairs in the donor and recipient language using the Cartesian product of each word with each translated synonym. We remove any duplicates, and any pairs that also occur in the loanword dataset, as we do not want true positives labeled as negatives when training the loanword detection model.

To create the **hard negatives** dataset, we use the PanPhon package (Mortensen et al., 2016) to

[4] https://en.wikipedia.org/wiki/List_of_languages_by_writing_system

[5] One such example is Hindi अगर (/əgər/), which can be both a loan from Persian, meaning "if," and a descendent of Sanskrit अगरु, referring to a type of wood.

[6] https://www.omniglot.com/writing/finnish.htm

compute six edit distances (see Sec. 4.2) between the IPA transcriptions of the gathered loanwords, and up to 20,000 candidate lemmas of the donor language, which are also transliterated into the IPA using Epitran. The result here is that each loanword is paired with up to six candidates that have a low phonetic edit distance but are not the original word in the donor language. We remove duplicates where multiple distance metrics chose the same closest neighbor, and where pairs cooccur with the synonyms or loans datasets.

Finally in the **randoms** dataset, we pair each loan with a random word in the donor language.

4 Similarity Metrics

Every word pair in the WikLoW dataset has measures of textual, phonetic, semantic, and articulatory similarity associated with it.

4.1 Textual Similarity

This is simply the Levenshtein edit distance between two strings. Where the two languages are written with different scripts, this is simply the maximum length of the strings, but in some cases, a language written in the same script as the donor language may borrow a word and keep the spelling unchanged, even if the pronunciation changes. A case in point is the word "science," a loan derived from French *science*, which is spelled identically but pronounced very differently (/saɪən(t)s/ vs. /sjɑ̃s/). Textual edit distance may be a useful feature for some language pairs, so we keep this metric.

4.2 Phonetic Similarity

Having created IPA transcriptions of the words, we compute 6 distance metrics over the transcriptions, all available from the PanPhon package:

- **Fast Levenshtein Distance**. A C implementation of Levenshtein distance (Levenshtein et al., 1966). PanPhon sets all edit costs to 1.
- **Dolgo Prime Distance**. Based the notion of the Dolgopolsky list of the 15 most stable lexemes (Dolgopolsky, 1986) but extended by PanPhon to a list of 14 most stable phonemes. Phonemes are mapped to these classes, over which Levenshtein distance is calculated.
- **Feature Edit Distance**. IPA is converted to articulatory feature vectors (e.g., storing presence, absence, or irrelevance of articulatory features place/manner of articulation, roundedness, pulmonic quality, etc.). Levenshtein distance is calculated over the feature vectors.

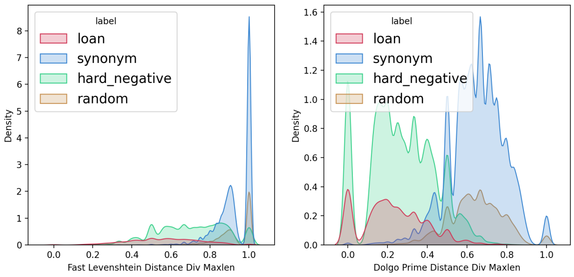

Figure 1: KDE plots of Fast Levenshtein and Dolgo Prime distances.

- **Hamming Feature Distance**. Same as Levenshtein distance, but with substitution cost being the Hamming distance (Hamming, 1950) between the feature vectors, normalized by the length of the vector.
- **Weighted Feature Distance**. Accounts for the class of the IPA symbol when calculating the Levenshtein costs as well as the probability of that specific edit. Weights are prespecified by PanPhon.
- **Partial Hamming Feature Distance**. Insertion and deletion costs are 1, however the cost of substitution for a zero value is half the substitution cost for a nonzero value.

We use the PanPhon normalized version of all edit distances, which divides by the maximum length of the two words in the pair. Fig. 1 shows kernel density estimation plots of the distribution of Fast Levenshtein and Dolgo Prime distances over the entire dataset. Loans have the lowest distance on average, followed by hard negatives.

4.3 Semantic Similarity

A loanword between a pair of languages must both sound and mean the same. While phonetic similarity, calculated with edit distance, has been a foundation for past work in loanword detection, modern large language models provide an opportunity to select for semantic similarity between word vectors, provided the models are trained over multilingual data. We make use of the simultaneous multilingual training objectives of MBERT (Devlin et al., 2019) and XLM (Conneau et al., 2020) to benefit from cross-language proximity of contextualized word embeddings, as shown in (Cao et al., 2019). We use the cosine function as our vector similarity measure.

MBERT is the multilingual version of BERT, pretrained on 104 languages, with demonstrated capacity for knowledge transfer on downstream tasks. It differs from BERT in two ways: i) in its masked language modeling pretraining, each batch comprises sentences from all languages, and ii) its dictionary is shared among all languages and is created by WordPiece from concatenating all corpora. Pires et al. (2019) show that MBERT's ability to transfer is due to a multilingual representation, which enables it to manage transfer across different scripts. These representations seem to share a common subspace that contains linguistic information, independent of specific languages.

XLM-100 is a cross-lingual (100-language) pretrained model which extends previous BERT-based models with a Translation Language Modeling (TLM) objective as well as the masked language and causal language modeling objectives, and has demonstrated success in unsupervised machine translation tasks (Conneau and Lample, 2019). XLM uses byte-pair encoding subword tokenization (Sennrich et al., 2016) which includes the most frequent symbol pairs when creating the token vocabulary. This makes it suitable for encoding tokens common in low-resourced languages (LRL) while alleviating bias towards high-resource languages, by reducing tokenization of LRL words at the character level. This improves the alignment of embedding spaces of languages that share either the same alphabet or proper nouns (Smith et al., 2017), both of which occur frequently among loanwords.

To these models, we input a "sentence" consisting of the word preceded by the `[CLS]` or `<bos>` token and followed by the `[SEP]`/`<eos>` token. We retrieve the vector of the `[CLS]`/`<bos>` token as a representation of the entire semantics of the input, to account for tokenization possibly splitting the word.

4.4 Alignment Network

To account for different phonotactics in paired languages (e.g., Swedish /skuːlɑː/→ Finnish /koulu/), we build a model to align phonemes in a word pair and account for epenthesis, elision, and metathesis, which provides a more informative measure than simply edit distance. Mortensen et al. (2016) show that information-rich phonological representations do better than character-based models or one-hot encodings in tasks such as NER.

We convert the IPA transcriptions to 21 subsegmental articulatory features using PanPhon[7]. These features were padded to the maximum length of a vector in the borrower-donor pair. The features for the loanword and original word were then concatenated for input to the alignment network.

The alignment network is a deep feedforward neural network trained on the aforementioned concatenated features of the *alldata* split of our datasets. The network was trained against the loan/non-loan binary label. This is not to predict loan status, but because we do not include any semantic information at this step, the label acts as an indicator of "phonetically aligned" or not. A positive prediction means the model predicts that the two words in the pair are strongly phonetically aligned according to the articulatory features. During inference, we get the pre-sigmoid logit value as a holistic alignment score between the two words.

5 Evaluation

For evaluation, we create three data distributions for each language pair. One (the **balanced** distribution), contains half loanwords and half non-loans. This is a well-behaved distribution well-suited for machine learning. The non-loans are drawn roughly $\frac{1}{7}$ from the hard negatives, $\frac{4}{7}$ from the synonyms, and $\frac{2}{7}$ from the randoms, reflecting the notion that relatively few words in a language are likely to be very phonetically close to a loanword on average, while there are likely to be many more words of synonymous or similar meaning.

Another distribution attempts to approximate the actual proportion of loanwords from the donor language into the recipient language (the "realistic" distribution, or **realdist**). Sometimes this proportion is well-documented, and at other times not.[8]. Where a figure is provided in the linguistic literature, we use it. Otherwise, we take the number of loanwords we collected from Wiktionary and divide it by the total number of lemmas in the borrowing language, and impose a lower bound of 10%, to maintain enough loanwords in the testing set. The non-loans portion of the *realdist* set is drawn in the same proportions as in the *balanced* set. For all language pairs currently in the WikLoW dataset, the *realdist* contains <50% loanwords, but for other language pairs, e.g., Korean-Chinese, >50% loanwords is certainly possible or likely (Sohn, 2005).

The final distribution (abbreviated **alldata**), takes all the data we collected from Wiktionary,

[7] Panphon does not contain suprasegmental or tonal information which may explain why alignment logits involving tonal languages such as Chinese may not sufficiently encode articulatory alignment (see Sec. 6)

[8] Sometimes documentation conflicts, such as Macrea (1961) and Sala (1988), which provide differing figures for Romanian loanwords from French, depending on whether all words or only core vocabulary is considered.

to purposely overweight the dataset against loanwords, to test our method in a difficult condition.

To each distribution, we concatenate two one-hot vectors representing the scripts of the languages in the pair. This allows certain models to learn dependencies between the scripts and other variables, e.g., if the languages are written in different scripts, the textual Levenshtein distance becomes nearly meaningless.

Each distribution was divided into a 90:10 train/test split, and then shuffled. We evaluate four different binary classifiers on all distributions: a logistic regressor (LR), a linear SVM, a Random Forest (RF), and a deep neural network (NN). The neural network consists of 3 layers of 512, 256, and 128 hidden units respectively, all with ReLU activation and followed by 10% dropout, and a final sigmoid activation, and is trained for 5,000 epochs with Adam optimization and BCE loss. We perform the evaluations listed below.

Single Multilingual Model (SMM) For each different data distribution, we train a single model on the data from every language pair listed in Table 1 except for *Persian-Arabic*, *Hungarian-German*, *German-Italian*, and *Catalan-Arabic*, which we reserve for subsequent experiments. The single multilingual model is evaluated on the unseen test sets for all language pairs used in training.

Pair-Specific Models For each distribution, we train and evaluate on a single language pair only, so we can compare the performance of the SMM to models specialized for each language pair.

Pruned Training Set We train on the *realdist* train set and evaluate on the *alldata* test set. This allows us to test on a much larger test set that contains a lower proportion of loanwords, and test the ability of our model to pick out loanwords from a more challenging distribution with less training data. The *realdist* train set is pruned of word pairs that appear in the *alldata* test set, since the two distributions were originally created separately. This experiment used the neural network classifier only.

Unseen Language Pairs We evaluate the performance of the SMM on *Persian-Arabic*, *Hungarian-German*, *German-Italian*, and *Catalan-Arabic*, which the model has *never* seen. This experiment used the neural network classifier only.

6 Results

Our primary metrics are precision, recall, and F1-score on positive loanword identification. Table 2 shows the average positive F1 score on the *realdist* distribution of the 4 classifiers we evaluated. The remaining tables and figures all focus on the results of the neural network, are sorted by decreasing number of loanwords in the language pair, and are discussed in Sec. 7. Table 3 presents the SMM results. Fig. 2 shows the *alldata* test results from Table 3 in bar graph form compared to the performance of the loanword detection model on each language pair when trained *only* on data from that language pair, and to the model when trained on the smaller pruned *realdist* training data. Table 4 shows the SMM's performance on the unseen language pairs, and Fig. 3 plots F1 score against the number of loanwords in each pair's test set.

	LR	NN	SVM	RF
F1 (+)	85	86	84	85

Table 2: Average F1 (+) of 4 classifiers (as %)

7 Discussion

We can quantitatively compare our approach to that of Mi et al. (2021), who report 75.35% average precision, 74.09% average recall, and 74.71% average F1 on loanword detection in Uyghur on borrowings from Russian, Arabic, Turkish, and Chinese. Our results are on different language pairs but are comparable to or exceed this, particularly if the testing set is balanced between loans and non-loans.

In Fig. 2, we can see that in most cases, the multilingual model outperforms the single-pair models on the same language pair on loanword retrieval, though this effect is most pronounced in language pairs with a higher density of loanwords. The model trained on the smaller pruned *realdist* data sees an appreciable drop in precision, but an equal or greater *increase* in loanword recall, and this effect is especially pronounced in pairs with fewer loanwords in the data overall, suggesting that training on a more realistic distribution may be advantageous when prioritizing reducing false negatives.

Fig. 3 shows the correlation between test set size and performance of the SMM (including unseen language pairs). There appears to be a strong correlation between the proportion of loanwords in a test set (as expected, a balanced set leads to optimal performance), but also the raw size of the test set itself. The model performs better on larger test sets, unseen or not, regardless of what data it was trained on. We speculate that this may be because when a borrowing language borrows a lot of words from a donor language, it does so at around the same time (e.g., English from Norman French), meaning

	all	*en-fr*	*en-de*	*id-nl*	*pl-fr*	*ro-fr*	*kk-ru*	*ro-hu*	*de-fr*	*hi-fa*	*fi-sv*	*az-ar*	*zh-en*
P (+)	92	96	90	96	90	94	93	88	94	94	85	85	81
	98	97	98	99	97	96	98	99	98	97	98	98	98
	83	89	84	85	82	82	86	76	86	81	78	69	71
R (+)	81	91	87	90	73	82	88	61	75	86	68	71	51
	98	99	99	99	97	99	100	93	99	99	98	98	93
	75	88	84	85	66	73	81	49	63	72	56	62	47
F1 (+)	86	93	89	93	81	88	91	72	83	90	75	70	62
	98	98	98	99	97	98	99	96	98	98	98	98	95
	79	89	84	85	73	77	83	60	73	76	65	65	57

Table 3: Single multilingual NN model results as % (1st row: *realdist*, 2nd row: *balanced*, 3rd row: *alldata*).

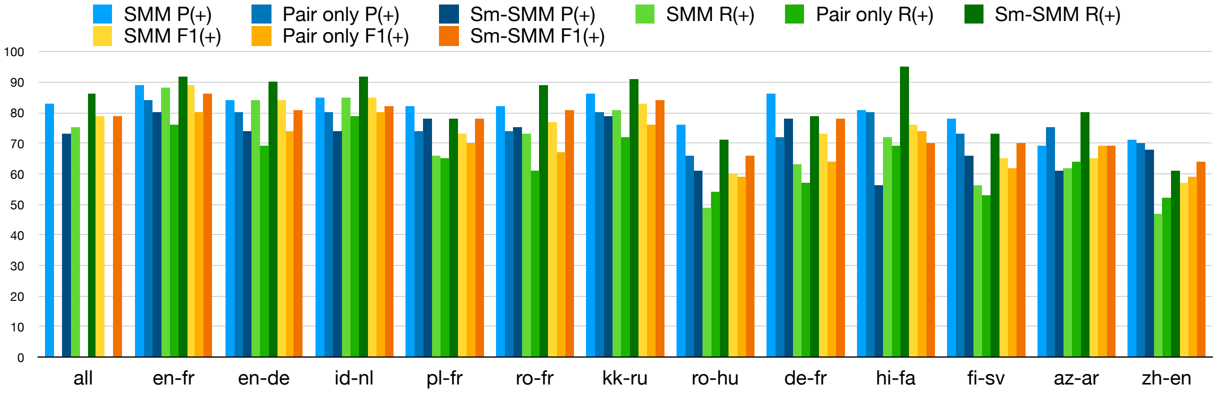

Figure 2: NN *alldata* results comparing base SMM, pair-specific model, and SMM trained on pruned *realdist* data (Small-SMM).

	fa-ar	*hu-de*	*de-it*	*ca-ar*
P (+)	95	95	73	100
	97	100	100	75
	75	73	54	25
R (+)	75	36	33	20
	97	93	92	30
	64	30	29	10
F1 (+)	84	52	46	33
	97	96	96	43
	69	43	38	14

Table 4: Holdout performance (same format as Table 3).

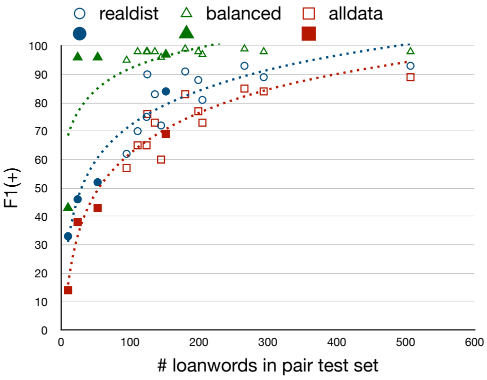

Figure 3: F1 score vs. number of loans per pair. Solid markers indicate unseen language pairs.

there are consistent transformations applied, which a network can pick up. This may not be the case in language pairs with a sparser density. Catalan-Arabic performance is particularly low and there are only 10 words in the test set, many of which were likely mediated by Spanish first.

7.1 Error Analysis

Mistakes made by the SMM, particularly on language pairs that perform less well, are illuminating.

Finnish-Swedish false negatives, e.g., *kyökki*/*kök* and *rontti*/*strunt*, suggest that additional final vowels and reduction of consonant clusters pose a difficulty. Other false negatives back this up, such as Mandarin-English 巧克力 (*qiǎokèlì*)/*chocolate* or Romanian-Hungarian *sudui*/*szidni*, which show sometimes irregular transformation to fit the borrowing language's phonotactics.

False positives are overwhelmingly hard negatives, and the model has particular trouble with languages that use abugidas or alphabets that borrow from languages that use abjads, due to the lack of vowels. Examples include Hindi-Persian निसार

(*nisār*)/*nasr* and Azerbaijani-Arabic *rəbb*/*rabbaba*. This can largely be attributed to Epitran not inserting vowels into Perso-Arabic transcriptions.

This suggests one clear way to potentially improve our method: incorporating multi-head attention into the phonemic alignment network rather than the current feedforward structure, which is performing the task the way single-head attention would and then averaging over all alignments.

Cognates are excluded from the positive loans data unless the cognate was actually later borrowed into the recipient language, as sometimes happens (e.g. "chef" vs. "head"). It is rare for cognates to be misclassified as loanwords due to intervening sound changes between two languages with common ancestry, but there are cases where a loanword is paired with a word in the source language that is cognate to it but is not the original borrowed word. Table 5 shows some of these rare cases.

Language pair	Word pair
en-fr	*communard*/*communal*
en-de	*Blume*/*Bluhm*
ro-fr	*cupolă*/*coupelle*
de-fr	*Montage*/*montant*

Table 5: Cognates mislabeled as loanwords by SMM.

7.2 Influence of Features

Neural networks are difficult to interpret, but the weights of the logistic regression classifier, which on average performed ~1-3% lower than the neural network, gives a sense of which features are important. Overall the alignment score is a strong positive correlate to loanword status across all language pairs. As expected, Levenshtein textual edit distance is inversely correlated with loanword status in pairs that share the same script, but not when the languages use different scripts. Interestingly, the semantic similarity metrics do not have a lot of influence on the model, but XLM is generally more influential than MBERT, and this influence is more pronounced among the lower-resourced languages (e.g., Kazakh-Russian, Hindi-Persian, Azerbaijani-Arabic), which supports XLM's claim to be more suited to LRLs, but the influence is most pronounced on English-French, the highest-resourced language pair currently in WikLoW, which undercuts the claim somewhat. Since loanwords are expected to be semantically similar, this task allows us to investigate the quality of multilingual language models on different language pairs. These findings are also borne out by ablation tests on the neural network classifier. For instance, dropping the alignment score and semantic similarities causes recall on the different-script pairs (Hindi-Persian, Azerbaijani-Arabic, Mandarin-English) to drop by 20% or more, while not affecting the same-script pairs as significantly. Sec. A.5 in the appendix shows these findings in more detail.

7.3 Human Comparison

To compare the performance of our model to human performance on loanword retrieval, we selected three language pairs, English-French, Hindi-Persian, and Mandarin-English, took the list of loanwords from the test set of the *alldata* distribution, and asked N annotators who were fluent speakers of each borrowing language to mark which in the list they thought were loans from the listed donor language. This was a fast way to assess human loanword recall and provide comparative numbers to our system on these language pairs. Table 6 shows the results.

Pair	N	Human μ R(+)	SMM R(+)	κ	# loans (homonyms)
en-fr	7	29	**88**	.059	508 (8)
hi-fa	6	60	**72**	.113	125 (4)
zh-en	6	**85**[9]	47	.034	95 (1)

Table 6: Human average loanword recall vs. SMM recall (as %).

Our system is able to significantly exceed human recall on English-French and Hindi-Persian, but not on Chinese-English (as noted those numbers may be inflated). Some loans were also homonyms, which may have had a small impact on human recall (see supplement). We also calculated Fleiss' kappa (Fleiss, 1971) over the human annotations and found that even when individual humans demonstrated moderate-to-high recall on loanword retrieval, there was virtually no agreement among annotators on which loanwords they identified.

8 Conclusions and Future Work

Automated loanword detection enables a number of downstream tasks. Coreferents and named entities across languages may often be loanwords, and common vocabulary enables potential improvements in machine translation (Ortega et al., 2021).

Parallel corpora of loanwords also afford learning cross-lingual contextual word embedding mappings—inspired by the success of pre-

[9] These numbers may be artificially high due to the Chinese annotators being bilingual in English.

Transformer embedding mappings (Bojanowski et al., 2016), and the potential of post-Transformer alignments (Cao et al., 2019). These can be incorporated into the Transformer architecture to provide auxiliary signals to enhance translation in two ways: i) Introducing another multi-head attention between the input language embeddings and their mappings in the target language space—similar to the second multi-head attention block in the original Transformer architecture (Vaswani et al., 2017). We propose to map embeddings between a source language L_X and target language L_Y by computing a transformation matrix between paired representations of semantically-equivalent words or sentences, then to compute attention weights between these mapped embeddings, and concatenate these auxiliary attention outputs with the attention between tokens from L_X and already-generated tokens from L_Y. ii) Unmasking identified loanwords in the target language in the decoder's input, which is expected to provide further context to the decoder in the target language. This would replicate a uniquely human linguistic capability: the ability to pick up context in an unfamiliar language by picking out known words (i.e., loans from a known language). Fig. 4 shows a proposed architecture for these operations.

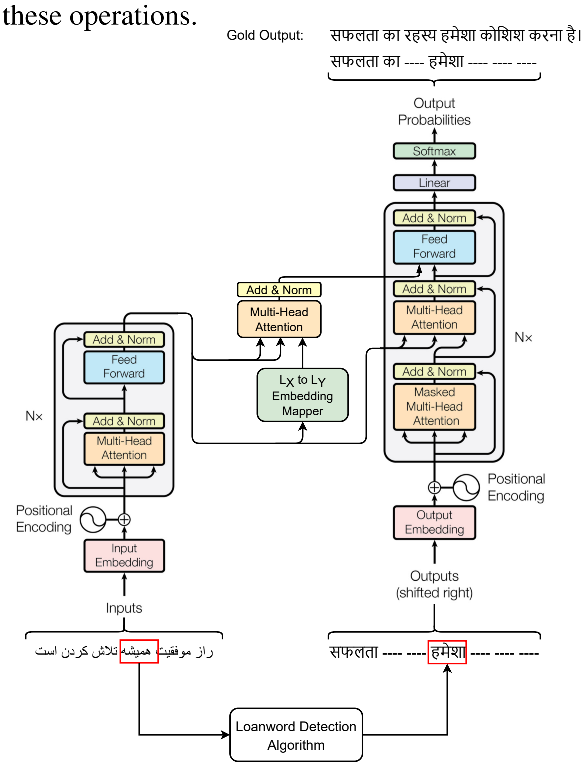

Figure 4: Proposed novel Transformer architecture for exploiting loanword knowledge in machine translation.

Mapping between embedding spaces also allows expanding our method and dataset to new languages not covered by MBERT or XLM through resources like IndicBERT (Kakwani et al., 2020).

8.1 Why Study Loanwords?

In keeping with the COLING 2022 special theme, "Tackling the Grand Challenges of the world by promoting mutual understanding through language," we posit that common vocabulary decreases barriers to communication, and representing it offers a particular benefit to LRLs in NLP, by providing a way to leverage resources from higher-resourced languages that have contributed vocabulary to an LRL. In this, Wiktionary itself has been and can continue to be a resource (Zesch et al., 2008; Krizhanovsky and Smirnov, 2013; De Melo, 2015; Wu and Yarowsky, 2020). Loanword detection is also necessarily not language agnostic, and is therefore important for linguistic diversity and inclusion in NLP (Joshi et al., 2020), although our multilingual results suggest that there may be key features of loanwords that allow detection to generalize.

We propose these challenges to the community:

1. We have presented a novel baseline for loanword detection across arbitrary language pairs that delivers high-quality results, but there remain challenges particularly for languages with divergent phonotactics.
2. We have also presented a method to gather more data for new languages, and demonstrated our detection method's performance on unseen language pairs, which we present as a baseline for comparison.
3. We have also provided homonym data, which is tailor-made to confound a loanword detection algorithm. Discriminating loanwords from their homonyms remains a challenge that presents many interesting opportunities in areas like machine translation and comparative and corpus linguistics.

Acknowledgments

Thanks to our human annotators: Iliana Castillon, Srishti Gupta, Ye (Christina) Jiang, Chirag Kandoi, Alex Karduna, Uma Krishnaswami, Vasantha Krishnaswami, Jacinda Li, Rachel Masters, Wenjun Ni, Evelyn Ochoa, Dhruva Patil, Aditya Raikwar, Ethan Seefried, Caspian Siebert, Videep Venkatesha, Xiaolu Wei, Tianyi Xiong, and Haodong Yang. Thanks to our reviewers for their helpful comments. We're sorry we couldn't get Arabic script to compile in LaTeX but it kept rendering backwards.

References

Elena Álvarez-Mellado and Constantine Lignos. 2022. Detecting Unassimilated Borrowings in Spanish: An Annotated Corpus and Approaches to Modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3868–3888.

Werner Betz. 1959. Lehnwörter und Lehnprägungen im vor-und Frühdeutschen. In *Band 1*, pages 135–164. de Gruyter.

Michael Bloodgood and Benjamin Strauss. 2017. Using Global Constraints and Reranking to Improve Cognates Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1983–1992, Vancouver, Canada. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.

Cecil H Brown, Eric W Holman, Søren Wichmann, and Viveka Velupillai. 2008. Automated classification of the world s languages: a description of the method and preliminary results. *Language Typology and Universals*, 61(4):285–308.

Steven Cao, Nikita Kitaev, and Dan Klein. 2019. Multilingual Alignment of Contextual Word Representations. In *Proceedings of the International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. *Cross-Lingual Language Model Pretraining*. Curran Associates Inc., Red Hook, NY, USA.

Alina Maria Cristea, Liviu P Dinu, Simona Georgescu, Mihnea-Lucian Mihai, and Ana Sabina Uban. 2021. Automatic Discrimination between Inherited and Borrowed Latin Words in Romance Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2845–2855.

Gerard De Melo. 2014. Etymological Wordnet: Tracing The History of Words. In *LREC*, pages 1148–1154. Citeseer.

Gerard De Melo. 2015. Wiktionary-based word embeddings. In *Proceedings of Machine Translation Summit XV: Papers*.

Marisa Delz. 2013. A theoretical approach to automatic loanword detection. *Master's thesis, Eberhard-Karls-Universität Tübingen*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Aaron B Dolgopolsky. 1986. A probabilistic hypothesis concerning the oldest relationships among the language families of northern Eurasia. *Typology, relationship and time: a collection of papers on language change and relationship by soviet linguists*, pages 27–50.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Richard W Hamming. 1950. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160.

Martin Haspelmath and Uri Tadmor. 2009. The loanword typology project and the world loanword database. *Loanwords in the world's languages: A comparative handbook*, pages 1–34.

Einar Haugen. 1950. The analysis of linguistic borrowing. *Language*, 26(2):210–231.

Gerhard Jäger. 2018. Global-scale phylogenetic linguistic inference from lexical resources. *Scientific Data*, 5(1):1–16.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Marisa Köllner. 2021. *Automatic Loanword Identification Using Tree Reconciliation*. Ph.D. thesis, Universität Tübingen.

Marisa Köllner and Johannes Dellert. 2016. Ancestral state reconstruction and loanword detection.

Grzegorz Kondrak. 2001. Identifying Cognates by Phonetic and Semantic Similarity. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Andrew A Krizhanovsky and Alexander V Smirnov. 2013. An approach to automated construction of a general-purpose lexical ontology based on Wiktionary. *Journal of Computer and Systems Sciences International*, 52(2):215–225.

Sofie Labat and Els Lefever. 2019. A Classification-Based Approach to Cognate Detection Combining Orthographic and Semantic Similarity Information. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 602–610, Varna, Bulgaria. INCOMA Ltd.

Els Lefever, Sofie Labat, and Pranaydeep Singh. 2020. Identifying Cognates in English-Dutch and French-Dutch by means of Orthographic Information and Cross-lingual Word Embeddings. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4096–4101, Marseille, France. European Language Resources Association.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Dimitrie Macrea. 1961. Originea și structura limbii române. *Probleme de lingvistică română*, pages 7–45.

Chenggang Mi, Lei Xie, and Yanning Zhang. 2020. Loanword identification in low-resource languages with minimal supervision. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(3):1–22.

Chenggang Mi, Yating Yang, Lei Wang, Xiao Li, and Kamali Dalielihan. 2014. Detection of loan words in Uyghur texts. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 103–112. Springer.

Chenggang Mi, Yating Yang, Lei Wang, Xi Zhou, and Tonghai Jiang. 2018. A neural network based model for loanword identification in Uyghur. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Chenggang Mi, Shaolin Zhu, and Rui Nie. 2021. Improving Loanword Identification in Low-Resource Language with Data Augmentation and Multiple Feature Fusion. *Computational Intelligence and Neuroscience*, 2021.

David R Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

David R Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484.

Andrea Mulloni and Viktor Pekar. 2006. Automatic Detection of Orthographics Cues for Cognate Recognition. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

John E Ortega, Richard Alexander Castro-Mamani, and Jaime Rafael Montoya Samame. 2021. Love Thy Neighbor: Combining Two Neighboring Low-Resource Languages for Translation. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*, pages 44–51.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Taraka Rama and Johann-Mattis List. 2019. An automated framework for fast cognate detection and bayesian phylogenetic inference in computational historical linguistics. Association for Computational Linguistics.

Benoît Sagot. 2017. Extracting an etymological database from Wiktionary. In *Electronic Lexicography in the 21st century (eLex 2017)*, pages 716–728.

Marius Sala. 1988. *Vocabularul reprezentativ al limbilor romanice*. Ed. Ştiinţifiča si Enciclopedică.

Charles Schafer and David Yarowsky. 2002. Inducing Translation Lexicons via Diverse Similarity Measures and Bridge Languages. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv e-prints*, page arXiv:1702.03859.

Ho-min Sohn. 2005. *Korean language in culture and society*. University of Hawaii press.

Hiroya Takamura, Ryo Nagata, and Yoshifumi Kawasaki. 2017. Analyzing semantic change in Japanese loanwords. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1195–1204.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Yulia Tsvetkov and Chris Dyer. 2015. Lexicon stratification for translating out-of-vocabulary words. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 125–131.

René Van Der Ark, Philippe Mennecier, John Nerbonne, and Franz Manni. 2007. Preliminary identification of language groups and loan words in central asia. In *Proceedings of the RANLP Workshop on Acquisition and Management of Multilingual Lexicons*, pages 13–20.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Uriel Weinreich. 1954. Languages in contact. In *Languages in Contact*. De Gruyter Mouton.

Huiyu Wu and Diego Klabjan. 2021. Logit-based Uncertainty Measure in Classification. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 948–956. IEEE.

Winston Wu and David Yarowsky. 2020. Wiktionary normalization of translations and morphological information. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4683–4692.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Using Wiktionary for Computing Semantic Relatedness. In *AAAI*, volume 8, pages 861–866.

Liqin Zhang, Franz Manni, Ray Fabri, and John Nerbonne. 2021. Detecting loan words computationally. *Variation Rolls the Dice. A Worldwide Collage in Honour of Salikoko S. Mufwene*.

Appendix

A.1 Further Details on Data Collection

We use the MediaWiki API to conduct our data collection. To maintain adherence to Wiktionary's terms of service, we make no more than 200 requests per second and sleep after a specified number of words are processed (by default, 200).

When conducting the initial data collection, we exclude terms that begin or end with hyphens, as those are likely to be affixes; that are only one letter long, as those are likely to contribute too much noise to the final dataset; and those that contain numerals or non-phonetic, non-syllabic, or non-logographic (depending on the language) symbols.

The choice of language pairs investigated here was determined in part by the intersection of languages that are supported by all 3 of Epitran, MBERT, and XLM-100, and that have a `[Recipient]_terms_borrowed_from_` `[Donor]` category on Wiktionary that contains more than 1,000 entries. The exceptions to this are: Finnish-Swedish, where Finnish is not natively supported by Epitran, but we built our own Finnish G2P mapping for Epitran; Mandarin-English, where some terms were discarded during preprocessing, causing the number to fall below 1,000; and Hungarian-German, German-Italian, and Catalan-Arabic, which were selected specifically for having fewer than 1,000 loanwords listed in Wiktionary.

Table 7 shows the 2-letter ISO 639-1 codes for these languages, which can help in interpreting Table 3 (Sec. 6).

A.2 Further Details on Semantic Similarity

In our experiments, for the XLM-100 and MBERT models, we extract the <`bos`> embeddings (equivalent to the `[CLS]` token for MBERT) for a word pair from the `last_hidden_state`. Numerous studies like (Jawahar et al., 2019) and (Tenney et al., 2019) suggest that BERT's later layers encode comparatively more high-level semantic information than its middle layers which tend to capture more syntactic features in the linguistic hierarchy. For both the models, the dimensions of the generated embeddings are of the shape (`batch_size`, `sequence_length`, `hidden_size`) where `batch_size` is 8 for both, `sequence_length` is the number of tokens from the word after tokenization (`max_length` is 512 for both models) whereas the embedding dimension i.e., `hidden_size` is 1280 for the XLM and 768 for MBERT. We then get the cosine similarities between the generated embeddings of each word pair of the borrower-donor pair in order to extract their semantic similarities.

Code	Language
ar	Arabic
az	Azerbaijani
ca	Catalan
de	German
en	English
fa	Persian
fi	Finnish
fr	French
hi	Hindi
hu	Hungarian
id	Indonesian
it	Italian
kk	Kazakh
nl	Dutch
pl	Polish
ro	Romanian
ru	Russian
sv	Swedish
zh	Mandarin

Table 7: ISO 639-1 language codes for languages in current dataset.

A.3 Further Details on Alignment Network

The alignment network was trained for 5,000 epochs with Binary Cross-Entropy (BCE) loss and Adam optimization, with a 20 percent validation set to prevent overfitting. The DNN consists of two hidden layers with 512 neurons each with ReLU activation, followed by 10% dropout, and an output layer and a sigmoid function.

Previous studies like Wu and Klabjan (2021) have suggested that logit outputs of neural networks can be a reliable and agnostic uncertainty measure that captures innate features of classes during classification and detection tasks. The alignment network here maps the concatenated articulatory features of a word pair to their class and therefore, the logits will contain class-based information that can subsequently be used as crucial features for our classifiers. In other words, these logits encode alignment information of the articulatory features that can be mapped to whether a pair is a phonetically similar, conditioned upon the sound patterns of their respective languages, or not.

A.4 Results from Other Classifiers

The main paper presented the results of the neural network classifier in detail and discussion of the weights from the logistic regressor. Here we present results from the logistic regression classifier (Table 8), the support vector machine (Table 9), and the random forest (Table 10).

The neural network is consistently the best-performing classifier, by about 1-5% F1, depending on which distribution is being evaluated on. The other classifiers can be expected to perform about this much lower. One thing to note is that the effect is most pronounced on the *alldata* dataset, which is the hardest dataset for any classifier on average, due to the overwhelming preponderance of non-loans. When the dataset is balanced between loans and non-loans, the type of classifier chosen for loanword detection is almost immaterial, with almost perfect performance all around. It seems at these proportions, the information encoded in the datasets, such as alignment score, edit distances, and cosine similarities, are informative enough. For this reason we have focused most discussion in the main body of the paper on the *alldata* and *realdist* datasets.

However, while the behavior of the logistic regressor and SVM are largely consistent with each other, and track that 1-5% difference with the neural network across all language pairs, the behavior of the random forest is rather different and inconsistent with the other classifiers. For example, it gets 100% recall on the balanced distributions of Indonesian-Dutch and Romanian-French (as well as Kazakh-Russian like the other classifiers), but on the Chinese-English *alldata* distribution, recall comes in ∼20% below the other classifiers. The other pairs with dissimilar scripts see a similar, albeit reduced effect on the same distribution, but so do some pairs that share a script, such as Indonesian-Dutch and Romanian-French.

A.5 Further Details on Influence of Features

This section contains the quantitative breakdown of the influence of different features on the results, which was discussed in Sec. 7.2. Fig. 5 is a graph representation of the logistic regressor weights mentioned there. The circular markers represent language pairs where both languages use the same script (including extended versions), while the square markers represent pairs where the languages use different scripts.

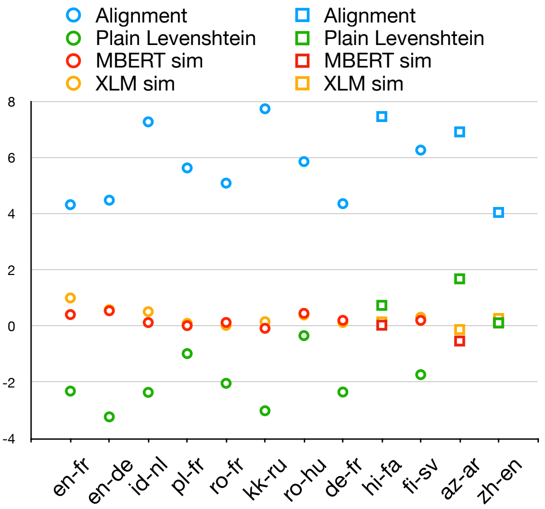

Figure 5: Logistic regressor weights trained on the *alldata* distribution.

Inferences drawn from the logistic regressor weights are bolstered by ablation tests on the neural network. Table 11 shows the neural network performance when the alignment scores and cosine similarities are not used as input features.

Articulatory alignment scores and cosine similarities are most important when the languages in the pair use different scripts. When these are removed as training inputs, and only phonetic and textual distance metrics are left, along with the script encodings, performance on the Azerbaijani-Arabic *alldata* distribution drops by 10% positive F1 and Hindi-Persian drops by 20% positive F1. The most drastic case is Mandarin-English, where without these features, positive F1 on *realdist* and *alldata* drop by 19% and 47% respectively, and positive recall drops by **20%** and **42%** respectively. This is because the different scripts make textual Levenshtein distance a useless feature here, and the differing phonologies of Mandarin and English make the phonetic edit distances noisy (e.g., see Sec. 7.1). Meanwhile, on certain same-script pairs, particularly those where words tend to be imported with little change in spelling (e.g., English-French, English-German, German-French), performance can actually go *up* slightly, because in these cases, textual Levenshtein distance is enough to detect that the word is a loan.

We should note that with only phonetic and script features, performance on the *balanced* distribution remains relatively high but suffers slightly. However, results vary on the *realdist* distribution,

	all	*en-fr*	*en-de*	*id-nl*	*pl-fr*	*ro-fr*	*kk-ru*	*ro-hu*	*de-fr*	*hi-fa*	*fi-sv*	*az-ar*	*zh-en*
P (+)	91	96	89	96	90	94	93	83	93	94	83	82	75
	98	97	97	98	97	96	99	98	98	97	95	97	97
	80	86	81	84	77	78	83	78	84	77	73	62	65
R (+)	80	88	83	89	76	82	86	66	73	83	69	65	52
	98	98	98	99	98	99	100	94	96	99	99	95	94
	72	85	80	83	65	73	76	48	59	68	58	51	46
F1 (+)	85	92	86	93	82	88	89	73	82	88	75	72	61
	97	97	97	98	97	98	99	96	97	98	97	96	95
	76	86	80	84	71	76	88	60	80	72	65	56	54

Table 8: Single multilingual logistic regression classifier results as % (1st row: *realdist*, 2nd row: *balanced*, 3rd row: *alldata*).

	all	*en-fr*	*en-de*	*id-nl*	*pl-fr*	*ro-fr*	*kk-ru*	*ro-hu*	*de-fr*	*hi-fa*	*fi-sv*	*az-ar*	*zh-en*
P (+)	92	96	89	97	90	94	93	85	93	93	82	80	75
	97	97	97	99	97	96	99	99	98	97	95	98	98
	80	86	79	83	77	78	82	78	85	80	74	62	66
R (+)	78	87	79	88	74	81	85	63	69	80	63	62	48
	98	98	98	99	98	99	100	94	97	99	99	95	94
	70	83	75	83	62	71	76	48	57	66	54	54	45
F1 (+)	84	92	84	92	81	87	89	72	79	86	71	70	59
	98	97	97	99	97	98	99	96	97	98	97	96	96
	75	85	77	83	69	74	79	60	68	73	62	58	54

Table 9: Single multilingual SVM classifier results as % (1st row: *realdist*, 2nd row: *balanced*, 3rd row: *alldata*).

	all	*en-fr*	*en-de*	*id-nl*	*pl-fr*	*ro-fr*	*kk-ru*	*ro-hu*	*de-fr*	*hi-fa*	*fi-sv*	*az-ar*	*zh-en*
P (+)	92	96	88	97	88	93	92	85	90	95	84	88	84
	96	96	95	95	96	93	97	95	97	97	94	99	100
	85	91	87	85	83	82	85	77	81	85	81	73	69
R (+)	80	90	84	89	76	80	91	73	71	77	65	59	40
	99	99	99	100	98	100	100	95	99	99	98	97	92
	68	86	82	77	58	62	78	47	54	59	54	43	26
F1 (+)	85	93	86	93	82	86	91	79	80	85	74	71	54
	97	97	97	97	97	96	98	95	98	98	96	98	96
	75	88	84	81	68	71	81	58	65	70	65	54	38

Table 10: Single multilingual random forest classifier results as % (1st row: *realdist*, 2nd row: *balanced*, 3rd row: *alldata*).

and there appears to be some correlation between increased performance on *realdist* without these features, and the proportion of loans in that distribution, suggesting that this is potentially important to consider (i.e., the base rate of loans from French into English, for instance, is relatively high). The performance penalty we see on LRLs and different-script pairs do suggest that overall the alignment score is most critical to generalizable performance, and the semantic similarities provide a way to analyze the quality of large multilingual language models for certain language pairs. These could also be augmented with other pair-specific metrics, such as overall measures of lexical or phonetic distance.

A.6 Homonyms in Human Comparison Task

The loanwords from the *alldata* test sets given to human annotators, that are also homonyms, are

	all	*en-fr*	*en-de*	*id-nl*	*pl-fr*	*ro-fr*	*kk-ru*	*ro-hu*	*de-fr*	*hi-fa*	*fi-sv*	*az-ar*	*zh-en*
P (+)	88	95	84	93	82	92	92	79	87	87	77	78	74
	94	95	96	92	97	90	99	95	97	92	93	91	81
	84	92	87	78	80	81	90	76	92	65	74	81	62
R (+)	87	95	97	92	67	93	95	66	96	94	85	73	31
	96	99	99	96	89	99	98	85	97	96	96	96	86
	72	92	93	73	46	77	93	44	93	48	67	41	5
F1 (+)	87	95	90	92	74	93	94	72	91	90	80	75	43
	95	97	97	94	93	94	99	90	97	94	94	94	84
	78	92	90	76	58	79	92	56	85	56	70	55	10

Table 11: Single multilingual NN classifier (without alignment and cosine similarity inputs) results as % (1st row: *realdist*, 2nd row: *balanced*, 3rd row: *alldata*).

listed below:

- English-French:
  - “punt,” from French *pointe*, meaning a bet or wager, with many other etymologies, including from Old English for a pontoon boat.
  - “Lemans,” French surname from toponym *Le Mans*, and from Middle English *Lemans*, “son of Leman.”
  - “bride,” from French *bride*, meaning a bridle, and from Old English *brȳd*, “bride, daughter-in-law.”
  - “paillard,” from a French surname (and name of a restaurant), and variant of “palliard,” meaning a beggar.
  - “lisse,” from French *lisser*, smooth, and from Old English *lissīan*, “to relieve.”
  - “tarse,” from French *tarse*, the tarsus or ankle-bones, and from archaic term for a male falcon.
  - “par,” from French *par*, meaning “through, by,” with many other etymologies, including from Latin *pār*, “equal.”
  - “bombard,” actually a doublet, with two meanings both meaning “cannon,” both ultimately from Middle French, one via modern French *bombarde*, the other via Middle English *bombard* (latter form also referred to a bassoon).

- Hindi-Persian:
  - अगर (*agar*), from Persian, meaning “if," and a descendent of Sanskrit अगरु (*agaru*), a type of wood.
  - देह (*deh*), from Persian, meaning “village," and a descendent of Sanskrit देह (*deha*), body.
  - मर्द (*mard*), from Persian, meaning “man," and a descendent of Sanskrit मर्द (*marda*), “destroying”.
  - हम (*ham*), from Persian, meaning “also," and natively Hindi ultimately from Sanskrit अस्मे (*asme*) , meaning “we,” “us.”

- Mandarin-English:
  - 塞特 (*sàitè*), from English *setter* but also from Hebrew male name *Seth*.

A.7 Proportion of Loanwords in Each Distribution

Table 12 shows the proportion of loanwords in each distribution for each language pair. The *balanced* distribution always contains 50% loans and so is not included.

A.8 Supported Languages and Scripts

Our system can in principle support the languages in Table 13 out of the box. While we have only tested on the language pairs mentioned in the main paper, and not every pairing in Table 13 has a sufficient volume of loanwords listed in Wiktionary, data collected in any of these languages can be converted to IPA with Epitran or extensions, and processed by MBERT and XLM to get cosine similarities between word vectors. Epitran can be extended to other languages by defining custom mapping, preprocessing, and postprocessing rules, as we did here for Finnish.

Proper functionality makes an assumption that the language given is written in the associated script

Pair	% Loans (*realdist*)	% Loans (*alldata*)
en-fr	30	14.841
en-de	10	13.322
id-nl	40	14.996
pl-fr	10	12.252
ro-fr	30	11.999
kk-ru	10	11.807
fa-ar	40	11.289
ro-hu	10	10.788
de-fr	10	10.206
hi-fa	30	10.126
fi-sv	10	9.754
az-ar	15	9.324
zh-en	10	10.496
hu-de	10	6.155
de-it	10	3.344
ca-ar	10	1.291

Table 12: Proportion of loanwords per pair in each distribution

listed. This serves the purpose of not only maintaining support in Epitran but also in collecting clean data from Wiktionary, and in assigning the correct one-hot script encoding during training and evaluation.

A.9 Organization of Code/Data

README.md contains instructions to run the full pipeline. language-pairs.json is a JSON file containing information about the language pairs to make datasets for, including codes for Epitran and Google Translate and desired *realdist* proportion of loans. language-pairs-holdout.json is the same for language pairs to be included in the holdout test set and withheld from training. language-pairs-pipelinetest.json contains only Catalan-Arabic, which is a small sample and runs (relatively) quickly, in order to validate the pipeline. These JSON files drive most of the rest of the code.

supported_languages.txt contains the list of supported languages (cf. Table 13). epitran-extensions contains preprocessing, mapping, and postprocessing rules for new Epitran language. Currently this contains only Finnish, which only uses pre and map. To run Epitran for the new language, these would need to be moved into the corresponding folder in the Epitran distribution in Python's site-packages.

ISO code	Language	Script
sq	Albanian	Latin
ar	Arabic	Latin
az	Azerbaijani	Latin
bn	Bengali	Bengali
my	Burmese	Myanmar
ca	Catalan	Latin
zh	Chinese	Chinese
hr	Croatian	Latin
cs	Czech	Latin
nl	Dutch	Latin
en	English	Latin
fi	Finnish	Latin
fr	French	Latin
de	German	Latin
hi	Hindi	Devanagari
hu	Hungarian	Latin
id	Indonesian	Latin
it	Italian	Latin
jv	Javanese	Latin
kk	Kazakh	Cyrillic
ky	Kyrgyz	Cyrillic
ms	Malay	Latin
ml	Malayalam	Malayalam
mr	Marathi	Devanagari
fa	Persian	Arabic
pl	Polish	Latin
pt	Portuguese	Latin
pa	Punjabi	Gurmukhi
ro	Romanian	Latin
ru	Russian	Cyrillic
es	Spanish	Latin
sw	Swahili	Latin
sv	Swedish	Latin
ta	Tamil	Tamil
te	Telugu	Telugu
tr	Turkish	Latin
uk	Ukranian	Cyrillic
ur	Urdu	Arabic
uz	Uzbek	Latin
vi	Vietnamese	Latin

Table 13: Currently supported languages and scripts.

wiktionary-scraper-python contains the scrapers for initial data collection. Results are saved in results.

Datasets contains the code to make the four dataset splits with edit distances. Results are sorted by type split: Loans, Synonyms, Hard-Negatives, and Randoms. Data files contain word pairs, IPA transcriptions of each word, English translations of each word (for interpretability by a more general audience, and not used in training or evaluation—translations may be inaccurate due to shortcomings in the Google Translate model for the language in question). Note that calculating all the dataset splits for a language pair, particularly the hard negatives, may take a *very* long time, up to days, due to the number of passes through the data. This inefficiency is the main reason why only a subset of the 24K available Romanian-French loans are used in these experiments. Decreasing the time complexity of calculating the hard negatives while maintaining the quality of the output is the topic of ongoing research. production_train_test is the directory containing the datasets that will be used for final evaluation. These are sorted by language pair and then by evaluation distribution: alldata, balanced, and realdist. Datasets also contains the human annotation spreadsheets in folder human_annotation.

Classifiers contains the code to both train the alignment network for a language pair and get the logit alignment score for each word pair, and to get the cosine similarities from MBERT and XML. Datasets with logit and similarity values are resaved in production_train_test. Classifiers also contains the code to perform evaluation under all conditions mentioned in the main body of this paper.

torch_models contains a saved instance of the single multilingual model. Final_results contains the results from that model and others, which are reported in this paper. FleissKappa contains the code to calculate Fleiss' kappa score over the human annotations (found inside Datasets).