Bridging linguistic typology and multilingual machine translation with multi-view language representations

Arturo Oncevay **Barry Haddow** **Alexandra Birch**
School of Informatics, University of Edinburgh, Scotland
a.oncevay@ed.ac.uk

Abstract

Sparse language vectors from linguistic typology databases and learned embeddings from tasks like multilingual machine translation have been investigated in isolation, without analysing how they could benefit from each other's language characterisation. We propose to fuse both views using singular vector canonical correlation analysis and study what kind of information is induced from each source. By inferring typological features and language phylogenies, we observe that our representations embed typology and strengthen correlations with language relationships. We then take advantage of our multi-view language vector space for multilingual machine translation, where we achieve competitive overall translation accuracy in tasks that require information about language similarities, such as language clustering and ranking candidates for multilingual transfer. With our method, we can easily project and assess new languages without expensive retraining of massive multilingual or ranking models, which are major disadvantages of related approaches.

1 Introduction

Recent surveys consider linguistic typology as a potential source of knowledge to support multilingual natural language processing (NLP) tasks (O'Horan et al., 2016; Ponti et al., 2019). Linguistic typology studies language variation in terms of their functional processes (Comrie, 1989). Several typological knowledge bases (KB) have been crafted, from where we can extract categorical language features (Littell et al., 2017). Nevertheless, their sparsity and reduced coverage present a challenge for an end-to-end integration into NLP algorithms. For example, the World Atlas of Language Structure (WALS; Dryer and Haspelmath, 2013) encodes 143 features for 2,679 languages, but their mean coverage per language is barely around 14%.

Dense and data-driven language representations have emerged in response. They are computed from multilingual settings of language modelling (Östling and Tiedemann, 2017) and neural machine translation (NMT) (Malaviya et al., 2017). However, the language diversity in the corpus-based representations is limited. The language coverage could be broadened with other knowledge, such as that encoded in WALS, to distinguish even more language properties. Therefore, to obtain the best of both views (KB and task-learned) with minimal information loss, we project a shared space of discrete and continuous features using a variant of canonical correlation analysis (Raghu et al., 2017).

For our study, we fuse language-level embeddings from multilingual machine translation with syntactic features of WALS. We inspect how much typological knowledge is present by predicting features for new languages. Then, we infer language phylogenies and inspect whether specific relationships are induced from the task-learned vectors.

Furthermore, to demonstrate that our approach has practical benefits in NLP, we apply our language vectors in multilingual NMT with language clustering (Tan et al., 2019) and adapt the ranking of related languages for multilingual transfer (Lin et al., 2019). As a side outcome, we identify that there is an ideal setting to encode language relationships in language embeddings from NMT. Finally, we present a simple tool to allow everyone to fuse, extend and compare their own representations[1].

2 Multi-view language representations

Our primary goal is to fuse parallel representations of the same language in one shared space, and **canonical correlation analysis** (CCA) allows us to find a projection of two views for a given set of

[1]Once the paper is published, we will release the code at: https://github.com/aoncevay/multiview-langrep

data. With CCA, we look for linear combinations that maximise the correlation of the two sources in each coordinate iteratively (Hardoon et al., 2004). After training, we can apply the transformation learned on a new sample from any view to obtain a CCA-based language representation[2].

CCA considers all dimensions of the two views as equally important. However, our sources are potentially redundant: KB features are mostly one-hot-encoded, whereas task-learned ones inherit the high dimensionality of the embedding layer. Moreover, few samples and sparsity could make the convergence harder. For the redundancy issue, **singular value decomposition** (SVD) is an appealing alternative. With SVD, we factorise the source data matrix to compute the principal components and singular values. Furthermore, to deal with sparsity, we adopt a truncated SVD approximation, which is also known as latent semantic analysis in the context of linear dimensionality reduction for term-count matrices (Dumais, 2004).

The two-step transformation of SVD followed by CCA is called **singular vector canonical correlation analysis** (SVCCA; Raghu et al., 2017) in the context of understanding the representation learning throughout neural network layers. That being said, we use SVCCA to get language representations and not to inspect a neural architecture[3].

3 Methodology and research questions

To embed linguistic typology knowledge in dense representations for a broad set of languages, we employ SVCCA (§2) with the following sources:

KB view. We employ the language vectors from the URIEL and lang2vec database (Littell et al., 2017). Precisely, we work with the k-NN vectors of the Syntax feature class (U_S; 103 feats.), that are composed of binary features encoded from WALS (Dryer and Haspelmath, 2013).

(NMT) Learned view. Firstly, we exploit the NMT-learned embeddings from the Bible (L_B; 512 dim.) (Malaviya et al., 2017). Up to 731 entries are available in lang2vec that intersects with U_S. They were trained in a many-to-English NMT model with a pseudo-token identifying the source language at the beginning of every input sentence.

Secondly, we take the many-to-English language embeddings learned for the language clustering task on multilingual NMT (L_W; 256 dim.) (Tan et al., 2019), where they use 23 languages of the WIT^3 corpus (Cettolo et al., 2012).

One main difference for the latter is the use of factors in the architecture, meaning that the embedding of every input token was concatenated with the embedded pseudo-token that identifies the source language. The second difference is the neural architecture used to extract the embeddings: the former use a recurrent neural network, whereas the latter a small transformer model (Vaswani et al., 2017).

Finally, we train a new set of embeddings (L_T) that we extracted from the 53 languages of the TED corpus (many-to-English) processed by Qi et al. (2018), using the approach of Tan et al. (2019)[4].

What knowledge do we represent? Each source embeds specialised knowledge to assess language relatedness. The KB vectors can measure typological similarity, whereas task-learned embeddings correlates with other kinds of language relationships (e.g. genetic) (Bjerva et al., 2019b). To analyse whether each kind of knowledge is induced with SVCCA, we assess the tasks of typological feature prediction (§4) and reconstruction of a language phylogeny (§5).

What is the benefit for multilingual NMT (and NLP)? Language-level representations can evaluate the distance between languages in a vector space. We then can assess their applicability on multilingual NMT tasks that require guidance from language relationships. Therefore, language clustering and ranking related partner languages for (multilingual) transfer are our study cases (§6).

4 Prediction of typological features

An example of a typological feature is a word order specification, like whether the adjective is predominately placed before or after the noun (features #24 and #25 of U_S). Our task consists in predicting syntactic features (U_S) leaving one-language and

[2] With language representations, we refer to an annotated or unsupervised characterisation of a language itself (e.g. Spanish or English), and not to word or sentence-level representations, as it is used in the recent NLP literature.

[3] As the SVD step performs a dimensionality reduction while preserving the most explained variance as possible, we can consider two additional parameters: a threshold value in the [0.5,1.0] range with 0.05 incremental steps, for the explained variance ratio of each view. With a value equal to 1, we bypass SVD and compute CCA only. We then tuned all our following experiments (see Appendix C for details).

[4] We prefer to use factored embeddings over initial pseudo-tokens as we identified that there is a difference for encoding information about language similarity (see §7).

	one-language-out		one-family-out		
	Single	SVCCA	#F.	Single	SVCCA
L_B (Bible)	**72.77**	71.68	134	**72.15**	70.62
L_W (WIT-23)	81.27	**84.83**	12	79.49	**79.68**
L_T (TED-53)	77.96	**85.37**	18	76.36	**81.06**

Table 1: Avg. accuracy (↑) of typological feature prediction per NMT-learned and SVCCA(U_S,L_*) setting.

one-language-family out to control phylogenetic relationships (Bjerva et al., 2019a). Previous work has shown that task-learned embeddings are potential candidates to predict features of a linguistic typology KB (Malaviya et al., 2017), and our goal is to evaluate whether SVCCA can enhance the NMT-learned language embeddings with typological knowledge from their KB parallel view.

Experimental setup. We use a Logistic Regression classifier per U_S feature, which is trained with the NMT-learned or SVCCA representations in both one-language-out and one-language-family-out settings. For prediction, we use the original embedding or its SVCCA projection as inputs.

Results. In Table 1, we observe that SVCCA outperformed their NMT-learned counterparts for L_W and L_T, where the performance is significantly better for the one-language-out setting. In the case of L_B (with 731 entries), we notice that the overall performance drops, and the SVCCA transformation cannot improve it. We argue that a potential reason for the accuracy dropping is the method used to extract the NMT-learned embeddings (initial pseudo-token instead of factors: §7), which could diminishes the information embedded about each language, and consequently, impacts the SVCCA projection. In conclusion, we notice that specific typological knowledge is usually hard to learn in an unsupervised way, and fusing them with KB vectors using SVCCA is feasible for inducing information of linguistics typology in some scenarios.

5 Language phylogeny analysis

According to Bjerva et al. (2019b), there is a positive correlation between the language distances in a phylogenetic tree and a pairwise distance-matrix of task-learned representations. Our goal therefore is to investigate whether fusing linguistic typology with SVCCA can preserve or enhance the embedded relationship information. For that reason, we first examine how well a language phylogeny can be reconstructed from language representations (§5.1), and study the correlation afterwards (§5.2).

5.1 Inference of a phylogenetic tree

Experimental design. Based on previous work (Rabinovich et al., 2017), we take a tree of 17 Indo-European languages (Serva and Petroni, 2008) as a Gold Standard (GS), which is shown in Figure 1a. We also use agglomerative clustering with variance minimisation (Ward Jr, 1963) as linkage, but we employ cosine similarity as Bjerva et al. (2019b). We also consider a concatenation ($\oplus$) of the KB and NMT-learned views as a baseline.

It is essential to highlight that none of the NMT-learned and $\oplus$ vectors have all the 17 language entries of the GS. Therefore, we can quickly preview one of the significant advantages of the SVCCA vectors, as we are able to represent "unknown" languages using one of the views. The NMT-learned views lack English, since they were extracted from the source side of a many-to-English system, but we were able to project the KB English vectors into the shared space. In addition, we project other four languages (Swedish, Danish, Latvian, Lithuanian) to complete the L_W embeddings of Tan et al. (2019) and Latvian to complete our own L_T set.

Evaluation metric. We differ from previous studies and use a tree edit distance metric, which is defined as the minimum cost of transforming one tree into another by inserting, deleting or modifying (the label of) a node. Specifically, we used the All Path Tree Edit Distance algorithm (APTED; Pawlik and Augsten, 2015, 2016), a novel one for the task. We chose an edit-distance method as it is more transparent for assessing what is the degree of impact for a single change of linkage in the GS.

As we need to compare inferred pruned trees with different number of nodes, we propose a normalised version given by: $\mathrm{nAPTED} = \mathrm{APTED}\,/(|\,\mathrm{GS}\,| + |\tau|)$, where τ is the inferred tree, and $|.|$ indicates the number of nodes. The denominator then is the maximum cost possible of deleting all nodes of τ and inserting each GS node.

Results. Table 2 shows the results for all settings, where the single-view scores are meagre in most of the cases. For instance, the U_S inferred tree (Fig.1c) requires 30 editions to resemble the GS. The exception is L_T (Fig.1d), which requires half the editions, although it is incomplete.

We observe that the best absolute and normalised scores are obtained by fusing U_S and L_T with

	Single	$U_S \oplus L_*$	SVCCA(U_S,L_*)
U_S (Syntax)	30 / 0.45		
L_B (Bible)	35 / 0.54	27 / 0.42	23 / 0.34
L_W (WIT-23)	35 / 0.62	23 / 0.41	27 / 0.48
L_T (TED-53)	15 / 0.26	18 / 0.29	**10** / **0.15**

Table 2: APTED and nAPTED scores (↓) between the GS and inferred trees from all scenarios. NMT-learned and concatenation ($\oplus$) can only reconstruct pruned trees of 16 (L_B), 12 (L_W) and 15 (L_T) languages.

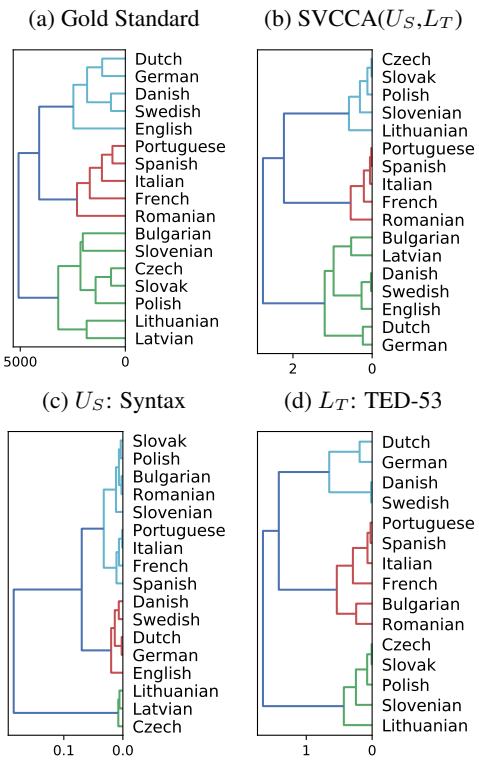


Figure 1: Gold Standard phylogeny (a) and reconstructed trees (b-d). L_T is smaller.

SVCCA (Fig.1b). English is projected in the Germanic branch, although Latvian is separated from the Balto-Slavic group. The latter case is similar for Bulgarian, which is misplaced in the original L_T tree as well. Nevertheless, we only require ten editions to equate the GS (where 66 is the maximum cost possible), confirming that our approach is a robust alternative for completing language entries and inferring a language phylogeny. We then proceed to discuss what kind of relationship we are representing.

5.2 Correlation with lexical similarity

Bjerva et al. (2019b) argued that raw language embeddings from language modelling correlates with *genetic* and structural similarity[5]. For the former, they correlated a distance matrix with pairwise-leaf-distances of the GS. However, Serva and Petroni (2008) originally inferred the phylogeny by comparing the translated Swadesh list of 200-words (Dyen et al., 1992) with Levenshtein (edit) distance. The list is a crafted set of concepts for comparative linguistics (e.g. I, eye, sleep), and it is usually processed by lexicostatistics methods to study language relationship through time. Therefore, we prefer to argue that corpus-based embeddings could partially encode **lexical** similarity of languages.

We perform an Spearman correlation between the cophenetic matrix[6] of the GS and the pairwise cosine-distance matrices of U_S, L_T and SVCCA(U_S,L_T), where we obtain correlation coefficients of 0.48, 0.68 and 0.80, respectively (p-values$<$0.001). Our conclusion is that typological knowledge strengthen the representation of lexical similarity within NMT-learned embeddings.

6 Application in multilingual NMT

With multilingual NMT, we can translate several language-pairs using a single model. Low-resource languages are usually benefited through multilingual transfer, which resembles a simultaneous training of the parent(s) and child models. Therefore, we want to take advantage of a language-level vector space for relating similar languages and enhancing multilingual transfer within multilingual NMT. For that reason, we first address the language clustering task proposed by Tan et al. (2019), and afterwards, the language ranking model of Lin et al. (2019).

Language clustering. The main idea is to obtain smaller multilingual NMT models as an intermediate point between maintaining many pairwise systems and a single massive multilingual model. With limited resources, it is challenging to support the first scenario, whereas the advantages for the massive setting are also very appealing (e.g. simplified training process, translation improvement for low-resource languages or zero-shot translation (Johnson et al., 2017)). Therefore, to address the task, Tan et al. (2019) trained a factored multilingual NMT model of 23 languages from Cettolo et al. (2012), where the language embedding is

[5] We note that Bjerva et al. (2019b) used monolingual texts translated from different languages to investigate what kind of genetic information is preserved. Concerning structural similarity, they computed a distance matrix using syntax-dependency-tags counts per language from annotated treebanks. We leave this analysis for further work.

[6] Pairwise-distances of the hierarchy's leaves (languages).

concatenated in every input token. Then, they performed hierarchical clustering with the representations, and selected a number of clusters guided by the Elbow method. Finally, they compared the systems against individual, massive and language family-based cluster models.

Rather than only using our multi-view representations to compute a set of clusters, we also address the question: do we need to train the massive model again if we want to add one or more new languages to our setting? If one of the goals of working with clustered NMT models is to avoid training and maintaining massive systems, it is quite a significant problem for the new language scenario.

Language ranking. The original goal of LANGRANK is to choose a parent language to perform transfer learning in different tasks, NMT included. To achieve this, Lin et al. (2019) trained a model based on the performance of several hundred pairwise MT systems using the dataset of Qi et al. (2018). For the input features, they considered linguistically-informed vectors from lang2vec (Littell et al., 2017) and corpus-based statistics, such as word/sub-word overlapping and the ratio of the token-types or the data size between the target child and potential candidates, where the latter features were one of the most relevant.

Considering the transfer capabilities within multilingual NMT and the possibility to obtain a ranked list of candidates from LANGRANK, we propose an adapted task of choosing k-related languages for multilingual transfer. We then use our multi-view representations to rank related languages from the vector space, as they embed information about typological and lexical relationships. This is similar to the features that Lin et al. (2019) considers, but without training a ranking model fed with scores from pairwise MT systems.

6.1 Experimental setup

We focus on the many-to-one (English) multilingual NMT setting to simplify the findings in both tasks. However, similar experiments could be performed in a one-to-many direction. Details about models, training and inference are described in Appendix B.

Dataset. We use the dataset processed and tokenised by Qi et al. (2018) of 53 languages (TED-53), from where we learned our L_T embeddings. We opted for TED-53 to better evaluate the extensibility of clusters and because it is also used to train the LANGRANK model. The list of languages, set sizes and other details are included in Appendix A. Before preprocessing the text, we drop any sentences from the training sets which overlap with any of the test sets. Since we are building many-to-English multilingual systems, this is important, as any such overlap will bias the results.

Clustering settings. We first list the baselines and our approaches, with the number of clusters/models between brackets:

1. Individual [53]: Pairwise model per language.
2. Massive [1]: A single model for all languages.
3. Language families [20]: Based on historical linguistics. We divide the 33 Indo-European languages into 7 branches. Moreover, 11 groups only have one language.
4. KB [3]: U_S (Syntax) tends to agglomerate large clusters (with 4-13-33 languages) and behaves similar to the massive model.
5. Learned [11]: We train a set of 53 factored embeddings (L_T) similar to Tan et al. (2019).
6. Concatenation [18]: $U_S \oplus L_T$
7. SVCCA-53 [10]: Multi-view representation with SVCCA composing both U_S and L_T vectors. Figure 2 shows the inferred hierarchy.
8. SVCCA-23 [10]: Similar to the previous setting, but we use the set of 23 language embeddings L_W instead (Tan et al., 2019), and project the 30 complementary languages with SVCCA(U_S,L_W).

With the last setting, we are interrogating whether SVCCA is a useful method for rapidly increasing the number of languages without retraining massive models given new entries that require their NMT-learned embeddings for clustering.

Similar to Tan et al. (2019), we use hierarchical agglomeration with average linkage and cosine similarity. However, we choose a different criterion for choosing the optimal number of clusters.

Selection of number of clusters. The Elbow criteria has been suggested for this purpose (Tan et al., 2019); however, as we can see in Figure 2, it might be ambiguous. Thus, we propose using a heuristic called Silhouette (Rousseeuw, 1987), which returns a score in the [-1,1] range. A sample cluster with a silhouette close to 1 indicates that it is cohesive and well-separated. With the average silhouette of all samples, we vary the number of cluster partitions, and look for the peak value.

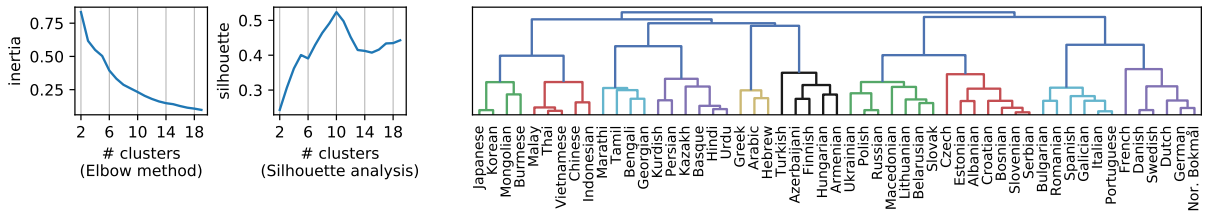


Figure 2: Clustering of TED-53 using the SVCCA-53 representations. At the left, we include the Elbow and Silhouette criteria to define the number of clusters. For the former, it is not clear what is the value to choose, whereas for the later we automatically select the highest peak at ten clusters.

Ranking settings. We focus on five low-resource languages from TED-53: Bosnian (bos, Indo-European/Balto-Slavic), Galician (glg, Indo-European/Italic), Malay (zlm, Austronesian), Estonian (est, Uralic) and Georgian (kat, Kartvelian). They have between 5k and 13k translated sentences with English, and we chose them as they achieved the most significant improvement from the individual to the massive setting. We then identified the top-3 related languages using LANGRANK, which give us a multilingual training set of around 500 thousand sentences for each case. Given that LANGRANK usually prefers to choose candidates with larger data size (Lin et al., 2019), for a fair comparison, we use SVCCA and cosine similarity to choose the k closest languages that can agglomerate a similar amount of parallel sentences.

6.2 Language clustering results

We first briefly discuss the composition of clusters obtained by SVCCA. Then, we analyse the results grouped by training size bins. We complement the analysis by family groups in Appendix D.

Cluster composition: In Figure 2, we observe that SVCCA-53 has adopted ten clusters with a proportionally distributed number of languages (the smallest one is Greek-Arabic-Hebrew, and the largest one has seven entries). Moreover, the languages are usually grouped by phylogenetic or geographical criteria.

From a more detailed inspection, there are entries that do not correspond to their respective family branches, although the single-view sources might induce the bias. For instance, the L_T tree (Fig1d) "misplaced" Bulgarian within Italic languages. Nevertheless, the unexpected agglomerations rely on the features encoded in the KB or the NMT learning process, and we expect they can uncover surprising clusters to avoid isolating languages without close relatives (e.g. Basque, or even Japanese as the only Japonic member in the set).

Training size bins: We manually define the upper bounds of the bins as [10,75,175,215] thousands of training sentences, which results in groups composed by [14,14,13,12] languages. Figure 3 shows the box plots of BLEU from where we can analyse each distribution (mean, variance).

Throughout all the bins, we observe that both SVCCA-53 and SVCCA-23 accomplish a comparable accuracy with the best setting in each group. In other words, their clusters provide stable performance for both low or high-resource languages.

In the first bin of the smallest corpora, the Massive baseline and the large clusters of U_S barely surpass the SVCCA schemes. Nevertheless, SVCCA contributes a notable advantage if we want to train a multilingual NMT model for a specific low-resource language, and we do not have the resources for training a massive system. We further analyse this scenario in §6.3.

In the rightmost bin, for the highest resource languages, the Massive and U_S performed worse than SVCCA. Furthermore, we show a competitive accuracy for the Individual and Family approaches. The former's clusters have steady performance across most of the bins as well. Nevertheless, they double the number of clusters that we have in both SVCCA settings, and with more than half of the "clusters" having only one language.

Other approaches, like using the NMT-learned embeddings (L_T) as Tan et al. (2019) or the concatenation baseline, obtain similar translation results in the last three bins. However, we need to obtain the NMT-learned embeddings first in order to fulfil those methods (from a 53-languages massive model). Using SVCCA and a pre-trained smaller set of language embeddings is enough for projecting new representations, as we present with our SVCCA-23 approach.

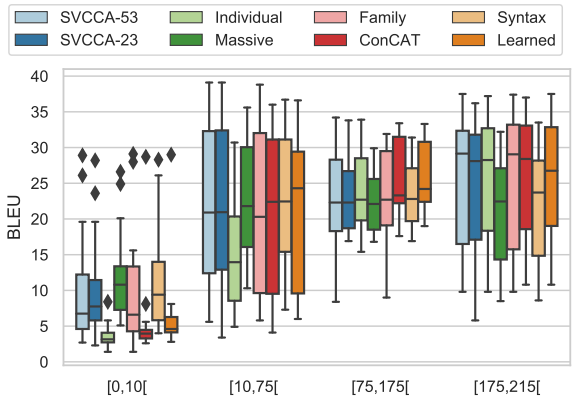


Figure 3: Box plots of BLEU scores per training-size bins. Each bin is represented by the range of minimum and maximum training size. Outliers are shown as diamonds.

6.3 Language ranking results

After discussing overall translation accuracy for all the languages, we now focus on five specific low-resource cases and how multilingual transfer enhance their performance. Table 3 shows the BLEU scores of the translation into English for the smaller multilingual models that group each child language with their candidates ranked by LANGRANK and our SVCCA-53 representations.

We also include the results of the individual and massive MT systems. Even when the latter baseline provides a significant improvement over the former, we observe that many of the smaller multilingual models outperform the translation accuracy of the massive system. The result suggests that the amount of data is not the most important confound for supporting multilingual transfer in a low-resource language.

Comparing the two ranking approaches, we observe that SVCCA achieves a comparable performance in most of the cases. We note that LANGRANK prefers related languages with large datasets, as it only requires three candidates to group around half a million training samples, whereas SVCCA suggests to include from three to ten languages to reach a similar amount of parallel sentences. However, increasing the number of languages could impact the multilingual transfer negatively (see the case of Georgian or *kat*), and it is analogous to adding different "out-of-domain" samples. To alleviate this, we could bypass candidate languages that do not possess a specific amount of training samples.

We argue that our method provides a robust alternative to determine which languages are suitable for multilingual transfer learning. A notable advantage is that we do not need to pre-train MT systems from a specific dataset, and we can easily extend the coverage of languages without re-training the ranking model to consider new entries[7].

L	Ind.	Mas.	LANGRANK	SVCCA-53
bos	4.2	26.6	28.8 (434)	28.2 (L=5)
glg	8.4	24.9	27.7 (443)	28.4 (L=3)
zlm	4.1	20.1	21.2 (463)	21.0 (L=4)
est	5.8	13.5	13.5 (533)	12.1 (L=6)
kat	5.8	14.3	13.3 (499)	10.5 (L=10)

Table 3: BLEU scores (L→English) for Individual, Massive and ranking approaches. LANGRANK shows the accumulated training size (in thousands) for the top-3 candidates, whereas with SVCCA we approximate the amount of data and include the number of languages.

7 Factors over initial pseudo-tokens

We additionally argue that the configuration used to compute the language embeddings impacts what relationship they can learn. For the analysis, we extract an alternative set of 53 language embeddings (L_{T*}) but using the initial pseudo-token setting instead of factors. Then, we perform a silhouette analysis to identify whether we can build cohesive and well-separated clusters of languages.

Figure 4 shows the silhouette analysis for the aforementioned embeddings (L_{T*}) together with the Bible embeddings (L_B) that were trained with the same configuration. We observe that the silhouette score never exceeds 0.2, and the curve keeps degrading when we examine a higher number of clusters, which contrast the trend shown in Figure 2. The pattern proves that the vectors are not suitable for clustering (the hierarchies are shown in Figure 6 in the Appendix), and they might only encode enough information to perform a classification task in the multilingual NMT training and inference. For that reason, we consider it essential to use language embeddings from factors for extracting language relationships.

[7]However, we do not answer what multilingual NMT really transfers to the low-resource languages. We left that question for further research, together with optimising the k number of languages or the amount of data per each language.

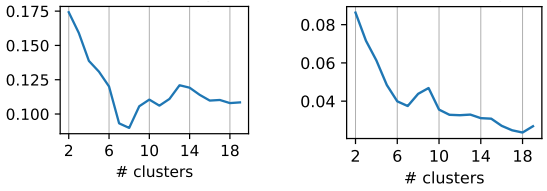


Figure 4: Silhouette analysis for the L_T* embeddings trained using an initial pseudo-token (left) and the L_B Bible vectors (right). Both cases present a downtrend curve with scores below 0.2. The hierarchies are shown in Figures 6d and 6e.

8 Related work

For language-level representations, URIEL and lang2vec (Littell et al., 2017) allow a straightforward extraction of typological binary features from different KBs. Murawaki (2015, 2017, 2018) exploits them to build latent language representations with independent binary variables. Language features are encoded from data-driven tasks as well, such as NMT (Malaviya et al., 2017) or language modelling (Tsvetkov et al., 2016; Östling and Tiedemann, 2017; Bjerva and Augenstein, 2018b) with complementary linguistic-related target tasks (Bjerva and Augenstein, 2018a).

Our approach is most similar to Bjerva et al. (2019a), as they build a generative model from typological features and use language embeddings, extracted from factored language modelling at character-level, as a prior of the model to extend the language coverage. However, our method primarily differs as it is mainly based in linear algebra, encodes information from both sources since the beginning, and can deal with a small number of shared entries (e.g. 23 from L_W) to compute robust representations.

There has been very little work on adopting typology knowledge for NMT. There is not a deep integration of the topics (Ponti et al., 2019), but one shallow and prominent case is the ranking method (Lin et al., 2019) that we analysed in §6.

Finally, CCA and its variants have been previously used to derive embeddings at word-level (Faruqui and Dyer, 2014; Dhillon et al., 2015; Osborne et al., 2016). Kudugunta et al. (2019) also used SVCCA but to inspect sentence-level representations, where they uncover relevant insights about language similarity that are aligned with our results in §5. However, as far as we know, this is the first time a CCA-based method has been used to compute language-level representations.

9 Takeaways and practical tool

We summarise our key findings as follows:

- SVCCA can fuse linguistic typology KB entries with NMT-learned embeddings without diminishing the originally encoded typological and lexical similarity of languages.
- Our method is a robust alternative to identify clusters and choose related languages for an small-scale multilingual transfer in NMT. The advantage is notable when it is not feasible to pre-train a ranking model or learn embeddings from a massive multilingual system.
- Factored language embeddings encodes more information to agglomerate related languages than the initial pseudo-token setting.

Furthermore, we release a tool to compute language representations using SVCCA, together with our L_T vectors. It is possible to use language vectors from KBs (e.g. lang2vec contains features from Phonology or Phonetic Inventory) or task-learned embeddings from different settings, such as one-to-many or many-to-many NMT and multilingual language modelling. Besides, we could rapidly project new language representations to assess tasks like clustering or ranking candidates for multilingual NMT (and NLP) that involves massive datasets of hundreds of languages.

10 Conclusion

We compute multi-view language representations with SVCCA using two sources: KB and NMT-learned vectors. We investigated that the knowledge contained in each source (typological and lexical similarity) is preserved in the combined representation. Moreover, our approach offers important advantages because we can evaluate projected languages with entries in only one of the views. The benefits are noticeable in multilingual NMT tasks, like language clustering and ranking related languages for multilingual transfer. We plan to study how to deeply incorporate our typologically-enriched embeddings in multilingual NMT, where there are promising avenues in parameter selection (Sachan and Neubig, 2018) and generation (Platanios et al., 2018).

Acknowledgments

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant

agreements No 825299 (GoURMET) and the EPSRC fellowship grant EP/S001271/1 (MT-Stretch). Also, it was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). We express our thanks to Kenneth Heafield, who provided us with access to the computing resources.

References

Antonios Anastasopoulos. 2019. A note on evaluating multilingual benchmarks.

Johannes Bjerva and Isabelle Augenstein. 2018a. From phonology to syntax: Unsupervised linguistic typology at different levels with language embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 907–916, New Orleans, Louisiana. Association for Computational Linguistics.

Johannes Bjerva and Isabelle Augenstein. 2018b. Tracking typological traits of uralic languages in distributed language representations. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 76–86, Helsinki, Finland. Association for Computational Linguistics.

Johannes Bjerva, Yova Kementchedjhieva, Ryan Cotterell, and Isabelle Augenstein. 2019a. A probabilistic generative model of linguistic typology. 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 1529–1540, Minneapolis, Minnesota. Association for Computational Linguistics.

Johannes Bjerva, Robert stling, Maria Han Veiga, Jrg Tiedemann, and Isabelle Augenstein. 2019b. What do language representations really represent? *Computational Linguistics*, 45(2):381–389.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Bernard Comrie. 1989. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Paramveer S Dhillon, Dean P Foster, and Lyle H Ungar. 2015. Eigenwords: Spectral word embeddings. *The Journal of Machine Learning Research*, 16(1):3035–3078.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Susan T Dumais. 2004. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230.

Isidore Dyen, Joseph B Kruskal, and Paul Black. 1992. An indoeuropean classification: A lexicostatistical experiment. *Transactions of the American Philosophical society*, 82(5):iii–132.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden. Association for Computational Linguistics.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535, Copenhagen, Denmark. Association for Computational Linguistics.

Yugo Murawaki. 2015. Continuous space representations of linguistic typology and their application to phylogenetic inference. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 324–334, Denver, Colorado. Association for Computational Linguistics.

Yugo Murawaki. 2017. Diachrony-aware induction of binary latent representations from typological features. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 451–461, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yugo Murawaki. 2018. Analyzing correlated evolution of multiple features using latent representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4371–4382, Brussels, Belgium. Association for Computational Linguistics.

John Nerbonne, Peter Kleiweg, Wilbert Heeringa, and Franz Manni. 2008. Projecting dialect distances to geography: Bootstrap clustering vs. noisy clustering. In *Data Analysis, Machine Learning and Applications*, pages 647–654, Berlin, Heidelberg. Springer Berlin Heidelberg.

Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2016. Survey on the use of typological information in natural language processing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1297–1308, Osaka, Japan. The COLING 2016 Organizing Committee.

Dominique Osborne, Shashi Narayan, and Shay B. Cohen. 2016. Encoding prior knowledge with eigenword embeddings. *Transactions of the Association for Computational Linguistics*, 4:417–430.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649, Valencia, Spain. Association for Computational Linguistics.

Mateusz Pawlik and Nikolaus Augsten. 2015. Efficient computation of the tree edit distance. *ACM Transactions on Database Systems (TODS)*, pages 3:1–3:40.

Mateusz Pawlik and Nikolaus Augsten. 2016. Tree edit distance: Robust and memory-efficient. *Information Systems*, 56:157–173.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 425–435, Brussels, Belgium. Association for Computational Linguistics.

Edoardo Maria Ponti, Helen OHoran, Yevgeni Berzak, Ivan Vuli, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Ella Rabinovich, Noam Ordan, and Shuly Wintner. 2017. Found in translation: Reconstructing phylogenetic language trees from translations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 530–540, Vancouver, Canada. Association for Computational Linguistics.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems 30*, pages 6076–6085. Curran Associates, Inc.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

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.

M. Serva and F. Petroni. 2008. Indo-European languages tree by Levenshtein distance. *EPL (Europhysics Letters)*, 81(6):68005.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao QIN, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1357–1366, San Diego, California. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244.

A Languages and individual BLEU scores

We work with 53 languages pre-processed by (Qi et al., 2018), from where we mapped the ISO 639-1 codes to the ISO 693-2 standard. However, we need to manually correct the mapping of some codes to identify the correct language vector in the URIEL (Littell et al., 2017) library:

- *zh* (*zho* , Chinese macro-language) mapped to *cmn* (Mandarin Chinese).
- *fa* (*fas* , Persian inclusive code for 11 dialects) mapped to *pes* (Western/Iranian Persian).
- *ar* (*ara* , Arabic) mapped to *arb* (Standard Arabic).

We disregard working with artificial languages like Esperanto (eo) or variants like Brazilian Portuguese (pt-br) and Canadian French (fr-ca).

Table 4 presents the list of all the languages with the following details: ISO 693-2 code, language family, size of the training set in thousands of sentences (with their respective training size bin) and the individual BLEU score obtained per clustering approach and other baselines.

B Model and training details

Similar to Tan et al. (2019), we train small transformer models (Vaswani et al., 2017). We jointly learn 90k shared sub-words with the byte pair encoding (Sennrich et al., 2016) algorithm built in SentencePiece (Kudo and Richardson, 2018). We also oversample all the training data of the less-resourced languages in each cluster, and shuffle them proportionally in all batches.

We use Nematus (Sennrich et al., 2017) only to extract the factored language embeddings from the TED-53 corpus (L_T). Given the large number of experiments, we choose the efficient Marian NMT (Junczys-Dowmunt et al., 2018) toolkit for training the rest of systems. With Marian NMT, we only use the basic pseudo-token setting for identifying the source language, as we did not need to retrieve new language embeddings after training. Besides, we allow the Marian NMT framework to automatically determine the mini-batch size given the sentence-length and available memory (mini-batch-fit parameter)

We train our models with up to four NVIDIA P100 GPUs using Adam optimiser (Kingma and

					BLEU score per approach							
ISO	Language	Lang. family	Size (k)	Bin	Individual	Massive	Family	U_S	L_T	$\oplus$	SVCCA-53	SVCCA-23
kaz	Kazakh	Turkic	3	1	2.5	5.3	4.0	4.3	3.3	2.7	3.3	3.0
bel	Belarusian	IE/Balto-Slavic	4	1	3.1	13.0	14.3	13.7	4.3	2.8	12.4	10.1
ben	Bengali	IE/Indo-Iranian	4	1	3.1	10.5	5.9	6.2	4.3	4.6	4.4	5.7
eus	Basque	Isolate	5	1	2.2	11.1	2.2	10.9	5.6	3.9	6.4	10.1
zlm	Malay	Austronesian	5	1	4.1	20.1	15.6	19.7	6.5	4.1	19.6	19.6
bos	Bosnian	IE/Balto-Slavic	5	1	4.2	26.6	28.0	28.3	6.5	4.1	26.1	23.6
urd	Urdu	IE/Indo-Iranian	5	1	3.9	11.8	7.5	8.0	5.5	5.6	7.1	6.8
aze	Azerbaijani	Turkic	5	1	2.8	8.1	6.4	6.7	4.2	3.2	7.3	7.4
tam	Tamil	Dravidian	6	1	1.4	5.1	1.4	4.0	2.8	2.6	2.7	2.3
mon	Mongolian	Mongolic	7	1	2.7	6.9	2.7	5.7	3.9	3.5	5.2	6.1
mar	Marathi	IE/Indo-Iranian	9	1	3.2	7.0	5.1	5.2	4.1	4.0	3.3	4.7
glg	Galician	IE/Italic	9	1	8.4	24.9	29.1	26.1	29.0	28.7	28.9	28.2
kur	Kurdish	IE/Indo-Iranian	10	1	4.0	10.1	6.8	10.8	4.9	3.6	6.3	8.1
est	Estonian	Uralic	10	1	5.8	13.5	10.5	14.1	8.1	8.1	11.7	11.9
kat	Georgian	Kartvelian	13	2	5.8	14.3	5.8	14.5	8.8	4.6	5.6	5.5
nob	Nor. Bokmal	IE/Germanic	15	2	19.0	35.2	38.8	36.4	35.0	35.0	39.1	39.1
hin	Hindi	IE/Indo-Iranian	18	2	8.1	16.0	8.8	10.5	9.5	6.2	8.3	8.6
slv	Slovenian	IE/Balto-Slavic	19	2	8.7	19.5	19.8	20.2	21.8	19.3	18.1	19.7
mya	Burmese	Sino-Tibetan	20	2	4.9	10.3	7.6	7.3	6.0	4.1	7.7	3.4
hye	Armenian	IE/Armenian	21	2	9.0	16.3	9.0	16.9	9.8	13.2	13.3	12.2
fin	Finnish	Uralic	23	2	8.5	14.4	11.5	14.9	8.3	8.3	12.1	15.0
mkd	Macedonian	IE/Balto-Slavic	24	2	15.7	26.8	27.3	27.4	27.2	28.0	25.1	22.6
lit	Lithuanian	IE/Balto-Slavic	41	2	12.2	17.9	19.4	18.4	20.0	19.0	17.9	18.6
sqi	Albanian	IE/Albanian	43	2	20.8	27.8	20.8	29.1	28.6	31.6	26.3	25.8
dan	Danish	IE/Germanic	44	2	30.7	35.6	38.4	36.7	34.4	34.4	38.9	39.0
por	Portuguese	IE/Italic	50	2	27.2	32.8	36.9	33.7	36.6	36.0	36.7	36.5
swe	Swedish	IE/Germanic	55	2	27.0	30.8	33.6	31.8	29.7	29.7	34.3	34.6
slk	Slovak	IE/Balto-Slavic	60	2	18.1	24.1	26.0	24.7	26.8	25.5	23.7	22.2
ind	Indonesian	Austronesian	85	3	23.8	24.3	21.4	26.0	28.0	27.0	26.5	26.5
tha	Thai	Kra-Dai	96	3	15.4	16.8	15.4	16.9	19.0	17.6	17.7	17.7
ces	Czech	IE/Balto-Slavic	101	3	20.7	22.1	23.9	22.8	24.2	23.3	21.2	22.1
ukr	Ukrainian	IE/Balto-Slavic	106	3	19.8	20.9	22.6	22.0	23.5	22.5	21.2	21.7
hrv	Croatian	IE/Balto-Slavic	120	3	28.5	27.5	30.4	28.9	30.8	31.5	28.3	26.7
ell	Greek	IE/Hellenic	132	3	31.9	29.9	31.9	30.9	32.2	33.4	34.2	32.7
srp	Serbian	IE/Balto-Slavic	134	3	26.4	25.6	28.3	27.1	28.8	29.4	26.3	25.4
hun	Hungarian	Uralic	145	3	19.1	17.2	17.0	17.9	21.3	17.7	18.0	18.7
fas	Persian	IE/Indo-Iranian	148	3	20.9	18.5	9.0	19.7	22.4	22.2	8.4	17.9
deu	German	IE/Germanic	165	3	30.1	25.5	29.5	26.9	31.4	31.7	29.9	29.6
vie	Vietnamese	Austroasiatic	169	3	22.7	20.3	22.7	21.6	23.6	22.2	22.3	22.3
bul	Bulgarian	IE/Balto-Slavic	172	3	33.9	29.9	31.9	31.4	33.3	33.1	34.2	33.8
pol	Polish	IE/Balto-Slavic	173	3	18.9	17.4	19.1	18.2	19.3	18.9	18.3	16.9
ron	Romanian	IE/Italic	178	4	30.0	25.8	30.7	27.0	28.1	30.8	30.8	29.6
tur	Turkish	Turkic	179	4	19.5	14.6	16.2	15.6	20.7	20.3	17.1	17.9
nld	Dutch	IE/Germanic	181	4	31.7	26.6	30.6	27.7	32.5	33.0	31.2	30.5
fra	French	IE/Italic	189	4	35.6	30.6	35.9	32.0	35.9	36.1	34.3	34.5
spa	Spanish	IE/Italic	193	4	37.2	32.2	37.4	33.5	37.5	37.0	37.5	36.2
cmn	Chinese	Sino-Tibetan	197	4	14.9	13.5	13.9	12.6	15.8	15.8	14.7	14.7
jpn	Japanese	Japonic	201	4	9.8	8.5	9.8	8.6	10.8	10.8	9.8	9.7
ita	Italian	IE/Italic	201	4	33.6	28.6	34.1	29.6	33.9	33.3	33.7	32.4
kor	Korean	Koreanic	202	4	14.4	12.2	14.4	11.9	15.1	15.0	13.3	5.8
rus	Russian	IE/Balto-Slavic	205	4	20.4	18.1	19.4	19.0	20.1	19.5	18.3	18.8
heb	Hebrew	Afroasiatic	208	4	32.4	24.4	32.9	25.8	29.9	30.3	31.9	31.6
arb	Arabic	Afroasiatic	211	4	26.5	20.5	27.5	21.6	25.4	26.5	27.5	26.6
			Average →		16.7	19.8	19.8	20.0	19.6	19.2	20.0	19.8

Table 4: List of languages with their BLEU scores per clustering approach (IE=Indo-European).

Ba, 2014) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.98, \varepsilon = 10^{-9}$) and early stopping at 5 validation steps for the cross-entropy metric. Finally, the sacreBLEU version string (Post, 2018) is as follows: BLEU+case.mixed+numrefs.1+smooth.exp +tok.13a+version.1.3.7.

C SVD explained variance selection

To compute SVCCA, we transform each source space using SVD, where we can choose to preserve a number of dimensions that represents an accumulated explained variance of the original dataset. For that reason, we perform a parameter sweep between 0.5 and 1.0 using 0.05 incremental steps. For a fair comparison, we also transform the single

Lang. families	# L	Size (k)	Individual	Massive	Family	U_S	L_T	$\oplus$	**SVCCA-53**	**SVCCA-23**
Isolate (Basque)	1	5	2.20	**11.10**	2.20	*10.90*	5.60	3.90	6.40 Δ-4.7	10.10 Δ-1.0
Dravidian	1	6	1.40	**5.10**	1.40	*4.00*	2.80	2.60	2.70 Δ-2.4	2.30 Δ-2.8
Mongolic	1	7	2.70	**6.90**	2.70	5.70	3.90	3.50	5.20 Δ-1.7	*6.10* Δ-0.8
Kartvelian	1	13	5.80	*14.30*	5.80	**14.50**	8.80	4.60	5.60 Δ-8.9	5.50 Δ-9.0
IE/Armenian	1	21	9.00	*16.30*	9.00	**16.90**	9.80	13.20	13.30 Δ-3.6	12.20 Δ-4.7
IE/Albanian	1	44	20.80	27.80	20.80	*29.10*	28.60	**31.60**	26.30 Δ-5.3	25.80 Δ-5.8
Kra-Dai	1	97	15.40	16.80	15.40	16.90	**19.00**	17.60	*17.70* Δ-1.3	*17.70* Δ-1.3
IE/Hellenic	1	132	31.90	29.90	31.90	30.90	32.20	*33.40*	**34.20**	32.70 Δ-1.5
Austroasiatic	1	170	*22.70*	20.30	*22.70*	21.60	**23.60**	22.20	22.30 Δ-1.3	22.30 Δ-1.3
Japonic	1	201	9.80	8.50	9.80	8.60	**10.80**	**10.80**	9.80 Δ-1.0	9.70 Δ-1.1
Koreanic	1	203	14.40	12.20	14.40	11.90	**15.10**	*15.00*	13.30 Δ-1.8	5.80 Δ-9.3
Austronesian	2	91	13.95	22.20	18.50	22.85	17.25	15.55	**23.05**	**23.05**
Sino-Tibetan	2	218	9.90	**11.90**	10.75	9.95	10.90	9.95	*11.20* Δ-0.7	9.05 Δ-2.8
Afroasiatic	2	420	29.45	22.45	**30.20**	23.70	27.65	28.40	*29.70* Δ-0.5	29.10 Δ-1.1
Uralic	3	180	11.13	15.03	13.00	**15.63**	12.57	11.37	13.93 Δ-1.7	*15.20* Δ-0.4
Turkic	3	189	8.27	9.33	8.87	8.87	*9.40*	8.73	9.23 Δ-0.2	**9.43**
IE/Germanic	5	462	27.70	30.74	34.18	31.90	32.60	32.76	**34.68**	*34.56* Δ-0.1
IE/Indo-Iranian	6	198	7.20	**12.32**	7.18	*10.07*	8.45	7.70	6.30 Δ-6.0	8.63 Δ-3.7
IE/Italic	6	823	28.67	29.15	**34.02**	30.32	33.50	*33.65*	*33.65* Δ-0.4	32.90 Δ-1.1
IE/Balto-Slavic	13	1,171	17.74	22.26	**23.88**	*23.24*	22.05	21.30	22.39 Δ-1.5	21.71 Δ-2.2
Weighted average $\rightarrow$			16.70	19.76	19.79	**20.03**	19.60	19.16	*19.97* Δ-0.1	19.82 Δ-0.1
Number of clusters/models $\rightarrow$			53	1	20	3	11	18	10	10

Table 5: BLEU score average per language family (IE=Indo-European). Every method includes the weighted BLEU average per number of languages (#L) and the number of clusters/models. Bold and italic represent first and second best results per family. Δ for SVCCA indicates the difference with respect to the highest score.

	Single	$U_S \oplus L_*$	SVCCA(U_S,L_*)
U_S (Syntax)	30 / 0.45 (0.5)		
L_B (Bible)	35 / 0.54 (0.9)	27 / 0.42 (0.70,0.55)	23 / 0.34 (0.70,0.75)
L_W (WIT-23)	35 / 0.62 (0.8)	23 / 0.41 (0.75,0.95)	27 / 0.48 (0.50,0.95)
L_T (TED-53)	15 / 0.26 (0.6)	18 / 0.29 (0.70,0.55)	**10 / 0.15** (1.00,0.55)

Table 6: Similar to Table 2, but including the optimal values for the SVD explained variance in each setting.

spaces (KB or Learned) with SVD and look for the optimal threshold.

Prediction of typological features. We selected a 0.5 threshold for the NMT-learned vectors of L_B and L_W, and 0.7 for L_T. In case of the SVCCA representation, L_T uses [0.75,0.70], whereas L_B and L_W employ [0.95,0.50] values. The parameter values are for both one-language-out and one-family-out settings. We can argue that there is redundancy in the NMT-learned embeddings, as the prediction of typological features with Logistic Regression always prefers a dimensionality-reduced version instead of the original data (threshold at 1.0).

Language phylogeny inference. In Table 6, we report the optimal value for the SVD explained variance ratio in each single and multi-view (concatenation and SVCCA) setting.

Language clustering (and ranking). We cannot perform an exhaustive analysis for the threshold of the explained variance ratio per view. As our main goal is to increase the coverage of languages steadily, we must determine what configuration allows a stable growth of the hierarchy.

We thereupon take inspiration from bootstrap clustering (Nerbonne et al., 2008), and increase the number of language entries from few entries (e.g. 10) to 53 by resample bootstrapping using each of the source vectors: U_S, L_T and L_W. Afterwards, we search for the threshold value that preserves a stable number of clusters given the peak silhouette value. Our heuristic looks for the least variability throughout the incremental bootstrapping (Fig. 5).

We found that 0.65 is the most stable value for U_S, whereas 0.60 is the best one for both L_T and L_W, so we thereupon fix SVCCA-53 and SVCCA-23 to [0.65,0.6]. We also apply the chosen thresholds on the concatenation baseline for a fair comparison. In the single-view cases, the transformations with the tuned variance ratio do not overcome any non-optimised counterparts.

D Language clustering results by language families

Following a guide for evaluating multilingual benchmarks (Anastasopoulos, 2019), we also group the scores by language families. Table 5 includes the overall weighted average per number of languages in each family branch. We observe that most of the approaches have obtained clusters with similar overall translation accuracy. The individual models are the only ones that significantly

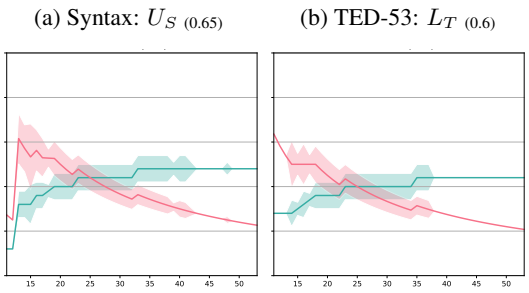


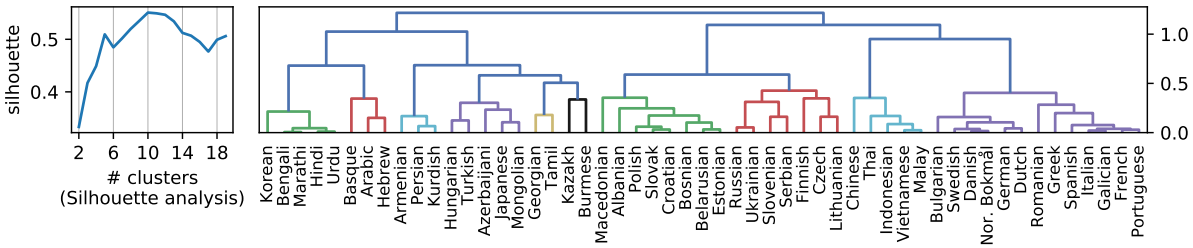
Figure 5: Analysis of the number of clusters (blue) and the ratio of number of clusters per total languages (red) given the chosen thresholds of explained variance ratio. We show the confidence interval computed from the bootstrapping, and we observe that the number of clusters is stable since 42 and 38 languages for U_S and L_T vectors, respectively.

underperform. The poor performance is transferred to the Family baseline, as most of the groups contains only one language given the low language diversity of the dataset.

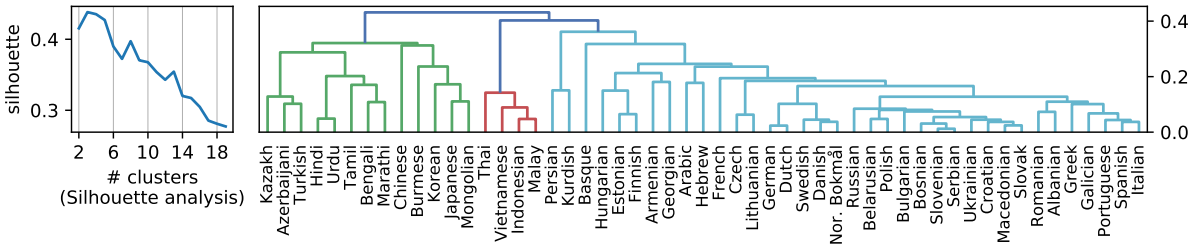
The U_S vectors obtain the highest overall accuracy, mostly from their few large clusters (see Fig. 6b). Meanwhile, SVCCA-53 achieves the second-best overall result, by a minimal margin, and with 3 to 7 languages per cluster, which are usually faster to converge. Besides, the massive model, the L_T embeddings and the concatenation baseline present a competitive achievement as well. However, the first requires more resources to train until convergence, whereas the last two need the 53 pre-trained embeddings from a previous massive system.

In contrast, SVCCA-23 is a faster alternative if we want to target specific new languages (see Fig. 6a). We only require a small group of language embeddings (e.g. L_W of 23 entries) and project the rest with SVCCA and a set KB-vectors as a side view. For instance, if we need to deploy a translation model for Basque or Thai, we could reach a comparable or better accuracy to a massive model with the SVCCA-23 chosen clusters of only 3 (Arabic, Hebrew) or 5 (Chinese, Indonesian, Vietnamese, Malay) languages, respectively.

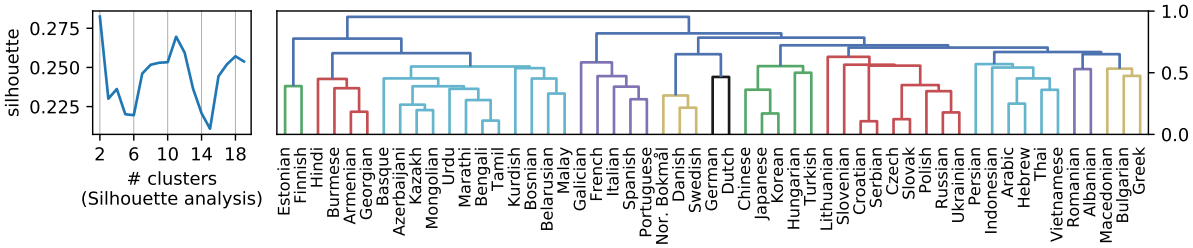
(a) SVCCA-23(U_S, L_W): SVCCA representations of Syntax and WIT-23



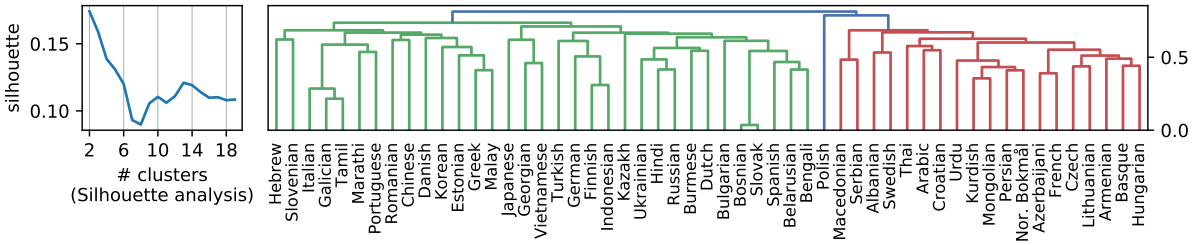
(b) U_S: Syntax



(c) L_T: NMT-learned from TED-53 (using factors)



(d) L_T*: NMT-learned from TED-53 but with initial pseudo-tokens



(e) L_B: NMT-learned from Bible

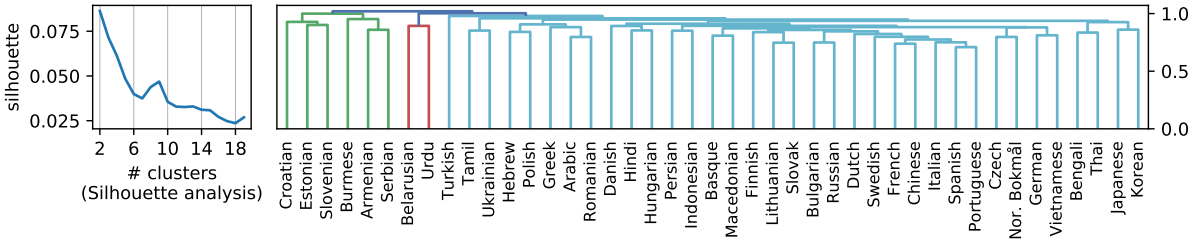


Figure 6: Silhouette analysis and dendrograms for clustering the 53 languages of TED-53 using different language representations. In (b), we observe that Syntax agglomerates a big cluster, similar to a massive approach. In (d) and (e), we note that the silhouette score is below 0.2 (1 is best), and the hierarchies do not define natural groups for the languages, as they are usually very separated from each other.