Mimicking Word Embeddings using Subword RNNs

Yuval Pinter **Robert Guthrie** **Jacob Eisenstein**

School of Interactive Computing
Georgia Institute of Technology
{uvp,rguthrie3,jacobe}@gatech.edu

Abstract

Word embeddings improve generalization over lexical features by placing each word in a lower-dimensional space, using distributional information obtained from unlabeled data. However, the effectiveness of word embeddings for downstream NLP tasks is limited by out-of-vocabulary (OOV) words, for which embeddings do not exist. In this paper, we present MIMICK, an approach to generating OOV word embeddings compositionally, by learning a function from spellings to distributional embeddings. Unlike prior work, MIMICK does not require re-training on the original word embedding corpus; instead, learning is performed at the type level. Intrinsic and extrinsic evaluations demonstrate the power of this simple approach. On 23 languages, MIMICK improves performance over a word-based baseline for tagging part-of-speech and morphosyntactic attributes. It is competitive with (and complementary to) a supervised character-based model in low-resource settings.

1 Introduction

One of the key advantages of word embeddings for natural language processing is that they enable generalization to words that are unseen in labeled training data, by embedding lexical features from large unlabeled datasets into a relatively low-dimensional Euclidean space. These low-dimensional embeddings are typically trained to capture distributional similarity, so that information can be shared among words that tend to appear in similar contexts.

However, it is not possible to enumerate the entire vocabulary of any language, and even large unlabeled datasets will miss terms that appear in later applications. The issue of how to handle these *out-of-vocabulary* (OOV) words poses challenges for embedding-based methods. These challenges are particularly acute when working with low-resource languages, where even unlabeled data may be difficult to obtain at scale. A typical solution is to abandon hope, by assigning a single OOV embedding to all terms that do not appear in the unlabeled data.

We approach this challenge from a quasi-generative perspective. Knowing nothing of a word except for its embedding and its written form, we attempt to learn the former from the latter. We train a recurrent neural network (RNN) on the character level with the embedding as the target, and use it later to predict vectors for OOV words in any downstream task. We call this model the MIMICK-RNN, for its ability to read a word's spelling and mimick its distributional embedding.

Through nearest-neighbor analysis, we show that vectors learned via this method capture both word-shape features and lexical features. As a result, we obtain reasonable near-neighbors for OOV abbreviations, names, novel compounds, and orthographic errors. Quantitative evaluation on the Stanford RareWord dataset (Luong et al., 2013) provides more evidence that these character-based embeddings capture word similarity for rare and unseen words.

As an extrinsic evaluation, we conduct experiments on joint prediction of part-of-speech tags and morphosyntactic attributes for a diverse set of 23 languages, as provided in the Universal Dependencies dataset (De Marneffe et al., 2014). Our model shows significant improvement

across the board against a single *UNK*-embedding backoff method, and obtains competitive results against a supervised character-embedding model, which is trained end-to-end on the target task. In low-resource settings, our approach is particularly effective, and is complementary to supervised character embeddings trained from labeled data. The MIMICK-RNN therefore provides a useful new tool for tagging tasks in settings where there is limited labeled data. Models and code are available at `www.github.com/yuvalpinter/mimick`.

2 Related Work

Compositional models for embedding rare and unseen words. Several studies make use of morphological or orthographic information when training word embeddings, enabling the prediction of embeddings for unseen words based on their internal structure. Botha and Blunsom (2014) compute word embeddings by summing over embeddings of the morphemes; Luong et al. (2013) construct a recursive neural network over each word's morphological parse; Bhatia et al. (2016) use morpheme embeddings as a prior distribution over probabilistic word embeddings. While morphology-based approaches make use of meaningful linguistic substructures, they struggle with names and foreign language words, which include out-of-vocabulary morphemes. Character-based approaches avoid these problems: for example, Kim et al. (2016) train a recurrent neural network over words, whose embeddings are constructed by convolution over character embeddings; Wieting et al. (2016) learn embeddings of character n-grams, and then sum them into word embeddings. In all of these cases, the model for composing embeddings of subword units into word embeddings is learned by optimizing an objective over a large unlabeled corpus. In contrast, our approach is a post-processing step that can be applied to any set of word embeddings, regardless of how they were trained. This is similar to the "retrofitting" approach of Faruqui et al. (2015), but rather than smoothing embeddings over a graph, we learn a function to build embeddings compositionally.

Supervised subword models. Another class of methods learn task-specific character-based word embeddings within end-to-end supervised systems. For example, Santos and Zadrozny (2014) build word embeddings by convolution over characters, and then perform part-of-speech (POS) tagging using a local classifier; the tagging objective drives the entire learning process. Ling et al. (2015) propose a multi-level long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997), in which word embeddings are built compositionally from an LSTM over characters, and then tagging is performed by an LSTM over words. Plank et al. (2016) show that concatenating a character-level or bit-level LSTM network to a word representation helps immensely in POS tagging. Because these methods learn from labeled data, they can cover only as much of the lexicon as appears in their labeled training sets. As we show, they struggle in several settings: low-resource languages, where labeled training data is scarce; morphologically rich languages, where the number of morphemes is large, or where the mapping from form to meaning is complex; and in Chinese, where the number of characters is orders of magnitude larger than in non-logographic scripts. Furthermore, supervised subword models can be combined with MIMICK, offering additive improvements.

Morphosyntactic attribute tagging. We evaluate our method on the task of tagging word tokens for their morphosyntactic attributes, such as gender, number, case, and tense. The task of morpho-syntactic tagging dates back at least to the mid 1990s (Oflazer and Kuruöz, 1994; Hajič and Hladká, 1998), and interest has been rejuvenated by the availability of large-scale multilingual morphosyntactic annotations through the Universal Dependencies (UD) corpus (De Marneffe et al., 2014). For example, Faruqui et al. (2016) propose a graph-based technique for propagating type-level morphological information across a lexicon, improving token-level morphosyntactic tagging in 11 languages, using an SVM tagger. In contrast, we apply a neural sequence labeling approach, inspired by the POS tagger of Plank et al. (2016).

3 MIMICK Word Embeddings

We approach the problem of out-of-vocabulary (OOV) embeddings as a **generation** problem: regardless of how the original embeddings were created, we assume there is a generative wordform-based protocol for creating these embeddings. By training a model over the existing vocabulary, we can later use that model for predicting the embedding of an unseen word.

Formally: given a language $\mathcal{L}$, a vocabulary $\mathcal{V} \subseteq \mathcal{L}$ of size V, and a pre-trained embeddings table $\mathcal{W} \in \mathbb{R}^{V \times d}$ where each word $\{w_k\}_{k=1}^{V}$ is assigned a vector $\boldsymbol{e}_k$ of dimension d, our model is trained to find the function $f : \mathcal{L} \rightarrow \mathbb{R}^d$ such that the projected function $f|_{\mathcal{V}}$ approximates the assignments $f(w_k) \approx \boldsymbol{e}_k$. Given such a model, a new word $w_{k^*} \in \mathcal{L} \setminus \mathcal{V}$ can now be assigned an embedding $\boldsymbol{e}_{k^*} = f(w_{k^*})$.

Our predictive function of choice is a **Word Type Character Bi-LSTM**. Given a word with character sequence $w = \{c_i\}_1^n$, a forward-LSTM and a backward-LSTM are run over the corresponding character embeddings sequence $\{\boldsymbol{e}_i^{(c)}\}_1^n$. Let $\boldsymbol{h}_f^n$ represent the final hidden vector for the forward-LSTM, and let $\boldsymbol{h}_b^0$ represent the final hidden vector for the backward-LSTM. The word embedding is computed by a multilayer perceptron:

$$f(w) = \mathbf{O}_T \cdot g(\mathbf{T}_h \cdot [\boldsymbol{h}_f^n; \boldsymbol{h}_b^0] + \boldsymbol{b}_h) + \boldsymbol{b}_T, \quad (1)$$

where $\mathbf{T}_h, \boldsymbol{b}_h$ and $\mathbf{O}_T, \boldsymbol{b}_T$ are parameters of affine transformations, and g is a nonlinear elementwise function. The model is presented in Figure 1.

The training objective is similar to that of Yin and Schütze (2016). We match the predicted embeddings $f(w_k)$ to the pre-trained word embeddings e_{w_k}, by minimizing the squared Euclidean distance,

$$\mathcal{L} = \|f(w_k) - \boldsymbol{e}_{w_k}\|_2^2 . \quad (2)$$

By backpropagating from this loss, it is possible to obtain local gradients with respect to the parameters of the LSTMs, the character embeddings, and the output model. The ultimate output of the training phase is the character embeddings matrix $\mathbf{C}$ and the parameters of the neural network: $\mathcal{M} = \{\mathbf{C}, \mathbf{F}, \mathbf{B}, \mathbf{T}_h, \boldsymbol{b}_h, \mathbf{O}_T, \boldsymbol{b}_T\}$, where $\mathbf{F}, \mathbf{B}$ are the forward and backward LSTM component parameters, respectively.

3.1 MIMICK Polyglot Embeddings

The pretrained embeddings we use in our experiments are obtained from Polyglot (Al-Rfou et al., 2013), a multilingual word embedding effort. Available for dozens of languages, each dataset contains 64-dimension embeddings for the 100,000 most frequent words in a language's training corpus (of variable size), as well as an *UNK* embedding to be used for OOV words. Even with this vocabulary size, querying words from respective UD corpora (train + dev + test) yields high

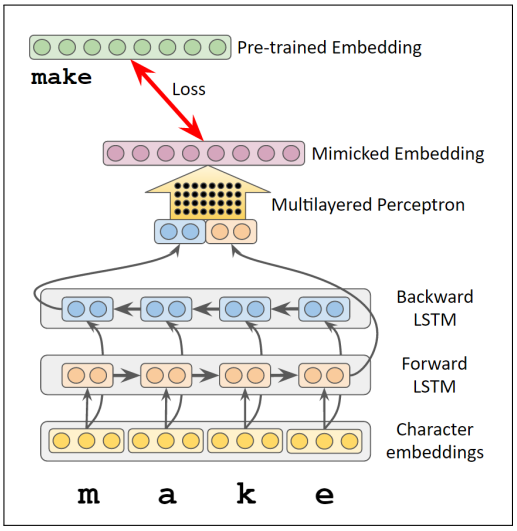


Figure 1: MIMICK model architecture.

OOV rates: in at least half of the 23 languages in our experiments (see Section 5), 29.1% or more of the word types do not appear in the Polyglot vocabulary. The token-level median rate is 9.2%.[1]

Applying our MIMICK algorithm to Polyglot embeddings, we obtain a prediction model for each of the 23 languages. Based on preliminary testing on randomly selected held-out development sets of 1% from each Polyglot vocabulary (with error calculated as in Equation 2), we set the following hyper-parameters for the remainder of the experiments: character embedding dimension = 20; one LSTM layer with 50 hidden units; 60 training epochs with no dropout; nonlinearity function $g = \tanh$.[2] We initialize character embeddings randomly, and use DyNet to implement the model (Neubig et al., 2017).

Nearest-neighbor examination. As a preliminary sanity check for the validity of our protocol, we examined nearest-neighbor samples in languages for which speakers were available: English, Hebrew, Tamil, and Spanish. Table 1 presents selected English OOV words with

[1] Some OOV counts, and resulting model performance, may be adversely affected by tokenization differences between Polyglot and UD. Notably, some languages such as Spanish, Hebrew and Italian exhibit **relational synthesis** wherein words of separate grammatical phrases are joined into one form (e.g. Spanish *del* = *de* + *el*, 'from the-masc.-sg.'). For these languages, the UD annotations adhere to the sub-token level, while Polyglot does not perform sub-tokenization. As this is a real-world difficulty facing users of out-of-the-box embeddings, we do not patch it over in our implementations or evaluation.

[2] Other settings, described below, were tuned on the supervised downstream tasks.

OOV word	Nearest neighbors	OOV word	Nearest neighbors
MCT	AWS OTA APT PDM SMP	compartmentalize	formalize rationalize discern prioritize validate
McNeally	Howlett Gaughan McCallum Blaney	pesky	euphoric disagreeable horrid ghastly horrifying
Vercellotti	Martinelli Marini Sabatini Antonelli	lawnmower	tradesman bookmaker postman hairdresser
Secretive	Routine Niche Turnaround Themed	developiong	compromising inflating shrinking straining
corssing	slicing swaying pounding grasping	hurtling	splashing pounding swaying slicing rubbing
flatfish	slimy jerky watery glassy wrinkle	expectedly	legitimately profoundly strangely energetically

Table 1: Nearest-neighbor examples for the English MIMICK model.

their nearest in-vocabulary Polyglot words computed by cosine similarity. These examples demonstrate several properties: (a) word shape is learned well (acronyms, capitalizations, suffixes); (b) the model shows robustness to typos (e.g., *develop**iong***, ***cor**ssing*); (c) part-of-speech is learned across multiple suffixes (*pesky – euphoric, ghastly*); (d) word compounding is detected (e.g., *lawnmower – bookmaker, postman*); (e) semantics are not learned well (as is to be expected from the lack of context in training), but there are surprises (e.g., *flatfish – slimy, watery*). Table 2 presents examples from Hebrew that show learned properties can be extended to nominal morphosyntactic attributes (gender, number – first two examples) and even relational syntactic subword forms such as genetive markers (third example). Names are learned (fourth example) despite the lack of casing in the script. Spanish examples exhibit word-shape and part-of-speech learning patterns with some loose semantics: for example, the plural adjective form *prenatales* is similar to other family-related plural adjectives such as *patrimoniales* and *generacionales*. Tamil displays some semantic similarities as well: e.g. *enjineer* ('engineer') predicts similarity to other professional terms such as *kalviyiyal* ('education'), *thozhilnutpa* ('technical'), and *iraanuva* ('military').

Stanford RareWords. The Stanford RareWord evaluation corpus (Luong et al., 2013) focuses on predicting word similarity between pairs involving low-frequency English words, predominantly ones with common morphological affixes. As these words are unlikely to be above the cutoff threshold for standard word embedding models, they emphasize the performance on OOV words.

For evaluation of our MIMICK model on the RareWord corpus, we trained the Variational Embeddings algorithm (VarEmbed; Bhatia et al., 2016) on a 20-million-token, 100,000-type Wikipedia corpus, obtaining 128-dimension word embeddings for all words in the test corpus. VarEmbed estimates a prior distribution over word embeddings, conditional on the morphological composition. For in-vocabulary words, a posterior is estimated from unlabeled data; for out-of-vocabulary words, the expected embedding can be obtained from the prior alone. In addition, we compare to FastText (Bojanowski et al., 2016), a high-vocabulary, high-dimensionality embedding benchmark.

The results, shown in Table 3, demonstrate that the MIMICK RNN recovers about half of the loss in performance incurred by the original Polyglot training model due to out-of-vocabulary words in the "All pairs" condition. MIMICK also outperforms VarEmbed. FastText can be considered an upper bound: with a vocabulary that is 25 times larger than the other models, it was missing words from only 44 pairs on this data.

4 Joint Tagging of Parts-of-Speech and Morphosyntactic Attributes

The Universal Dependencies (UD) scheme (De Marneffe et al., 2014) features a minimal set of 17 POS tags (Petrov et al., 2012) and supports tagging further language-specific features using attribute-specific inventories. For example, a verb in Turkish could be assigned a value for the evidentiality attribute, one which is absent from Danish. These additional morphosyntactic attributes are marked in the UD dataset as optional per-token attribute-value pairs.

Our approach for tagging morphosyntactic attributes is similar to the part-of-speech tagging model of Ling et al. (2015), who attach a projection layer to the output of a sentence-level bidirectional LSTM. We extend this approach to morphosyntactic tagging by duplicating this projection layer for each attribute type. The input to our multilayer perceptron (MLP) projection network is the hidden state produced for each token in the sentence by an underlying LSTM, and the output is

OOV word	Nearest neighbors
TTGFM '(s/y) will come true', GIAVMTRIIM 'geometric(m-pl)'$_2$	TPTVR '(s/y) will solve', TBTL '(s/y) will cancel', TSIR '(s/y) will remove' ANTVMIIM 'anatomic(m-pl)', GAVMTRIIM 'geometric(m-pl)'$_1$
BQFTNV 'our request'	IVFBIHM 'their(m) residents', XTAIHM 'their(m) sins', IRVFTV 'his inheritance'
RIC'RDSVN 'Richardson'	AVISTRK 'Eustrach', QMINQA 'Kaminka', GVLDNBRG 'Goldenberg'

Table 2: Nearest-neighbor examples for Hebrew (Transcriptions per Sima'an et al. (2001)). 's/y' stands for 'she/you-m.sg.'; subscripts denote alternative spellings, standard form being 'X'$_1$.

	Emb. dim	Vocab size	Polyglot in-vocab $N = 862$	All pairs $N = 2034$
VarEmbed	128	100K	41.9	25.5
Polyglot	64	100K	40.8	8.7
MIMICK	64	0	17.9	17.5
Polyglot +MIMICK	64	100K	40.8	27.0
Fasttext	300	2.51M		47.3

Table 3: Similarity results on the RareWord set, measured as Spearman's $\rho \times 100$. VarEmbed was trained on a 20-million token dataset, Polyglot on a 1.7B-token dataset.

attribute-specific probability distributions over the possible values for each attribute on each token in the sequence. Formally, for a given attribute a with possible values $v \in V_a$, the tagging probability for the i'th word in a sentence is given by:

$$\Pr(a_{w_i} = v) = (\text{Softmax}(\phi(\boldsymbol{h}_i)))_v \,, \quad (3)$$

with

$$\phi(\boldsymbol{h}_i) = \mathbf{O}^a_W \cdot \tanh(\mathbf{W}^a_h \cdot \boldsymbol{h}_i + \boldsymbol{b}^a_h) + \boldsymbol{b}^a_W, \quad (4)$$

where $\boldsymbol{h}_i$ is the i'th hidden state in the underlying LSTM, and $\phi(\boldsymbol{h}_i)$ is a two-layer feedforward neural network, with weights $\mathbf{W}^a_h$ and $\mathbf{O}^a_W$. We apply a softmax transformation to the output; the value at position v is then equal to the probability of attribute v applying to token w_i. The input to the underlying LSTM is a sequence of word embeddings, which are initialized to the Polyglot vectors when possible, and to MIMICK vectors when necessary. Alternative initializations are considered in the evaluation, as described in Section 5.2.

Each tagged attribute sequence (including POS tags) produces a loss equal to the sum of negative log probabilities of the true tags. One way to combine these losses is to simply compute the **sum loss**. However, many languages have large differences in sparsity across morpho-syntactic attributes, as apparent from Table 4 (rightmost column). We therefore also compute a **weighted sum loss**, in which each attribute is weighted by the proportion of training corpus tokens on which it is assigned a non-*NONE* value. Preliminary experiments on development set data were inconclusive across languages and training set sizes, and so we kept the simpler sum loss objective for the remainder of our study. In all cases, part-of-speech tagging was less accurate when learned jointly with morphosyntactic attributes. This may be because the attribute loss acts as POS-unrelated "noise" affecting the common LSTM layer and the word embeddings.

5 Experimental Settings

The morphological complexity and compositionality of words varies greatly across languages. While a morphologically-rich agglutinative language such as Hungarian contains words that carry many attributes as fully separable morphemes, a sentence in an analytic language such as Vietnamese may have not a single polymorphemic or inflected word in it. To see whether this property is influential on our MIMICK model and its performance in the downstream tagging task, we select languages that comprise a sample of multiple morphological patterns. Language family and script type are other potentially influential factors in an orthography-based approach such as ours, and so we vary along these parameters as well. We also considered language selection recommendations from de Lhoneux and Nivre (2016) and Schluter and Agić (2017).

As stated above, our approach is built on the Polyglot word embeddings. The intersection of the Polyglot embeddings and the UD dataset (version 1.4) yields 44 languages. Of these, many are under-annotated for morphosyntactic attributes; we select twenty-three sufficiently-tagged languages, with the exception of Indonesian.[3] Table 4 presents the selected languages and their typological properties. As an additional proxy for mor-

[3] Vietnamese has no attributes by design; it is a pure analytic language.

	Language	Branch	Script type	Morpho.	Tokens w/ attr.		Language	Branch	Script type	Morpho.	Tokens w/ attr.
vi	Vietnamese	Vietic	alphabetic*	Analytic	00.0%	fa	Persian	Iranian	consonantal	Agglutin.	65.4%
hu	Hungarian	Finno-Ugric	alphabetic	Agglutin.	83.6%	hi	Hindi	Indo-Aryan	alphasyllab.	Fusional	92.4%
id	Indonesian	Malayic	alphabetic	Agglutin.	—	lv	Latvian	Baltic	alphabetic	Fusional	69.2%
zh	Chinese	Sinitic	ideographic	Isolating	06.2%	el	Greek	Hellenic	alphabetic	Fusional	64.8%
tr	Turkish	Turkic	alphabetic	Agglutin.	68.4%	bg	Bulgarian	Slavic	alphabetic	Fusional	68.6%
kk	Kazakh	Turkic	alphabetic	Agglutin.	20.9%	ru	Russian	Slavic	alphabetic	Fusional	69.2%
ar	Arabic	Semitic	consonantal	Fusional	60.6%	cs	Czech	Slavic	alphabetic	Fusional	83.2%
he	Hebrew	Semitic	consonantal	Fusional	62.9%	es	Spanish	Romance	alphabetic	Fusional	67.1%
eu	Basque	Vasconic	alphabetic	Agglutin.	59.2%	it	Italian	Romance	alphabetic	Fusional	67.3%
ta	Tamil	Tamil	syllabic	Agglutin.	78.8%	ro	Romanian	Romance	alphabetic	Fusional	87.1%
						da	Danish	Germanic	alphabetic	Fusional	72.2%
						en	English	Germanic	alphabetic	Analytic	72.8%
						sv	Swedish	Germanic	alphabetic	Analytic	73.4%

Table 4: Languages used in tagging evaluation. Languages on the right are Indo-European. *In Vietnamese script, whitespace separates syllables rather than words.

phological expressiveness, the rightmost column shows the proportion of UD tokens which are annotated with any morphosyntactic attribute.

5.1 Metrics

As noted above, we use the UD datasets for testing our MIMICK algorithm on 23 languages[4] with the supplied train/dev/test division. We measure part-of-speech tagging by overall token-level accuracy.

For morphosyntactic attributes, there does not seem to be an agreed-upon metric for reporting performance. Dzeroski et al. (2000) report per-tag accuracies on a morphosyntactically tagged corpus of Slovene. Faruqui et al. (2016) report macro-averages of F1 scores of 11 languages from UD 1.1 for the various attributes (e.g., part-of-speech, case, gender, tense); recall and precision were calculated for the full set of each attribute's values, pooled together.[5] Agić et al. (2013) report separately on parts-of-speech and morphosyntactic attribute accuracies in Serbian and Croatian, as well as precision, recall, and F1 scores per tag. Georgiev et al. (2012) report token-level accuracy for exact all-attribute tags (e.g. 'Ncmsh' for "Noun short masculine singular definite") in Bulgarian, reaching a tagset of size 680. Müller et al. (2013) do the same for six other languages. We report **micro F1**: each token's value for each attribute is compared separately with the gold labeling, where a correct prediction is a matching non-*NONE* attribute/value assignment. Recall and precision are calculated over the entire set, with F1 defined as their harmonic mean.

[4]When several datasets are available for a language, we use the unmarked corpus.

[5]Details were clarified in personal communication with the authors.

5.2 Models

We implement and test the following models:

No-Char. Word embeddings are initialized from Polyglot models, with unseen words assigned the Polyglot-supplied *UNK* vector. Following tuning experiments on all languages with cased script, we found it beneficial to first back off to the lowercased form for an OOV word if its embedding exists, and only otherwise assign *UNK*.

MIMICK. Word embeddings are initialized from Polyglot, with OOV embeddings inferred from a MIMICK model (Section 3) trained on the Polyglot embeddings. Unlike the No-Char case, backing off to lowercased embeddings before using the MIMICK output did not yield conclusive benefits and thus we report results for the more straightforward no-backoff implementation.

CHAR→TAG. Word embeddings are initialized from Polyglot as in the No-Char model (with lowercase backoff), and appended with the output of a character-level LSTM updated during training (Plank et al., 2016). This additional module causes a threefold increase in training time.

Both. Word embeddings are initialized as in MIMICK, and appended with the CHAR→TAG LSTM.

Other models. Several non-Polyglot embedding models were examined, all performed substantially worse than Polyglot. Two of these

are notable: a random-initialization baseline, and a model initialized from FastText embeddings (tested on English). FastText supplies 300-dimension embeddings for 2.51 million lowercase-only forms, and no *UNK* vector.[6] Both of these embedding models were attempted with and without CHAR→TAG concatenation. Another model, initialized from only MIMICK output embeddings, performed well only on the language with smallest Polyglot training corpus (Latvian). A Polyglot model where OOVs were initialized using an averaged embedding of all Polyglot vectors, rather than the supplied *UNK* vector, performed worse than our No-Char baseline on a great majority of the languages.

Last, we do not employ type-based tagset restrictions. All tag inventories are computed from the training sets and each tag selection is performed over the full set.

5.3 Hyperparameters

Based on development set experiments, we set the following hyperparameters for all models on all languages: two LSTM layers of hidden size 128, MLP hidden layers of size equal to the number of each attribute's possible values; momentum stochastic gradient descent with 0.01 learning rate; 40 training epochs (80 for 5K settings) with a dropout rate of 0.5. The CHAR→TAG models use 20-dimension character embeddings and a single hidden layer of size 128.

6 Results

We report performance in both low-resource and full-resource settings. Low-resource training sets were obtained by randomly sampling training sentences, without replacement, until a predefined token limit was reached. We report the results on the full sets and on $N = 5000$ tokens in Table 5 (part-of-speech tagging accuracy) and Table 6 (morphosyntactic attribute tagging micro-F1). Results for additional training set sizes are shown in Figure 2; space constraints prevent us from showing figures for all languages.

MIMICK as OOV initialization. In nearly all experimental settings on both tasks, across languages and training corpus sizes, the MIMICK embeddings significantly improve over the Polyglot *UNK* embedding for OOV tokens on both POS and morphosyntactic tagging. For POS, the largest margins are in the Slavic languages (Russian, Czech, Bulgarian), where word order is relatively free and thus rich word representations are imperative. Chinese also exhibits impressive improvement across all settings, perhaps due to the large character inventory ($> 12{,}000$), for which a model such as MIMICK can learn well-informed embeddings using the large Polyglot vocabulary dataset, overcoming both word- and character-level sparsity in the UD corpus.[7] In morphosyntactic tagging, gains are apparent for Slavic languages and Chinese, but also for agglutinative languages — especially Tamil and Turkish — where the stable morpheme representation makes it easy for subword modeling to provide a type-level signal.[8] To examine the effects on Slavic and agglutinative languages in a more fine-grained view, we present results of multiple training-set size experiments for each model, averaged over five repetitions (with different corpus samples), in Figure 2.

MIMICK vs. CHAR→TAG. In several languages, the MIMICK algorithm fares better than the CHAR→TAG model on part-of-speech tagging in low-resource settings. Table 7 presents the POS tagging improvements that MIMICK achieves over the pre-trained Polyglot models, with and without CHAR→TAG concatenation, with 10,000 tokens of training data. We obtain statistically significant improvements in most languages, even when CHAR→TAG is included. These improvements are particularly substantial for test-set tokens outside the UD training set, as shown in the right two columns. While test set OOVs are a strength of the CHAR→TAG model (Plank et al., 2016), in many languages there are still considerable improvements to be obtained from the application of MIMICK initialization. This suggests that with limited training data, the end-to-end CHAR→TAG model is unable to learn a sufficiently accurate representational mapping from orthography.

7 Conclusion

We present a straightforward algorithm to infer OOV word embedding vectors from pre-trained,

[6] Vocabulary type-level coverage for the English UD corpus: 55.6% case-sensitive, 87.9% case-insensitive.

[7] Character coverage in Chinese Polyglot is surprisingly good: only eight characters from the UD dataset are unseen in Polyglot, across more than 10,000 unseen word types.

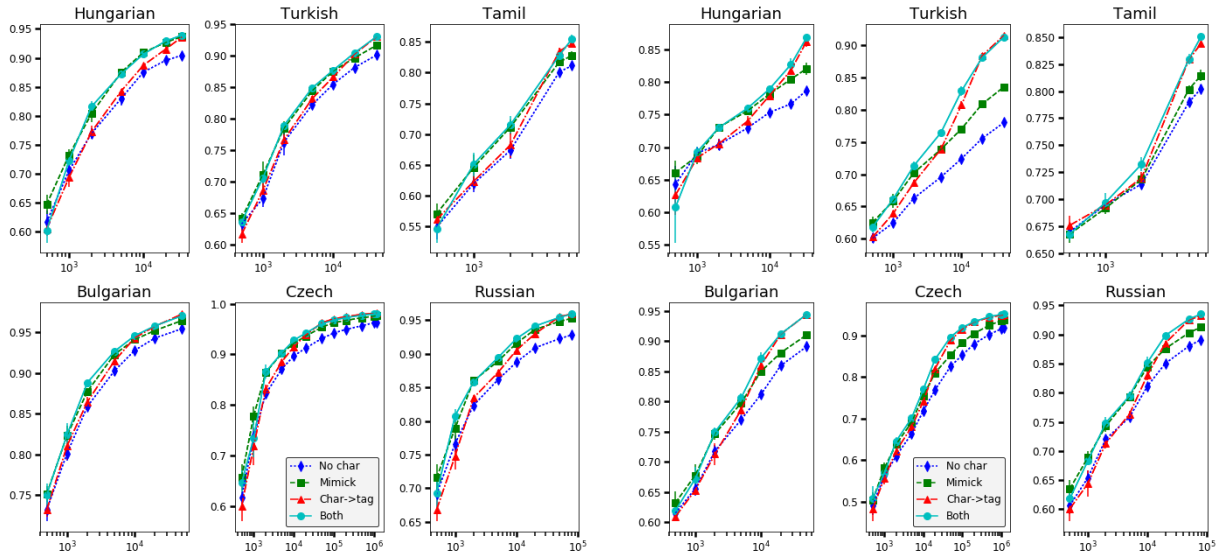
[8] Persian is officially classified as agglutinative but it is mostly so with respect to derivations. Its word-level inflections are rare and usually fusional.

	$N_{train} = 5000$				Full data					
	No-Char	MIMICK	CHAR →TAG	Both	N_{train}	No-Char	MIMICK	CHAR →TAG	Both	PSG 2016*
kk	—	—	—	—	4,949	81.94	83.95	83.64	84.88	
ta	82.30	81.55	84.97	85.22	6,329	80.44	**82.96**	84.11	84.46	
lv	80.44	**84.32**	84.49	**85.91**	13,781	85.77	**87.95**	89.55	89.99	
vi	85.67	*84.22*	84.85	85.43	31,800	89.94	90.34	90.50	90.19	
hu	82.88	**88.93**	85.83	**88.34**	33,017	91.52	**93.88**	94.07	93.74	
tr	83.69	**85.60**	84.23	**86.25**	41,748	90.19	**91.82**	93.11	92.68	
el	93.10	**93.63**	94.05	**94.64**	47,449	97.27	**98.08**	98.09	98.22	
bg	90.97	**93.16**	93.03	**93.52**	50,000	96.63	**97.29**	97.95	97.78	98.23
sv	90.87	**92.30**	92.27	**93.02**	66,645	95.26	**96.27**	96.69	96.87	96.60
eu	82.67	**84.44**	86.01	**86.93**	72,974	91.67	**93.16**	94.46	94.29	95.38
ru	87.40	**89.72**	88.65	**90.91**	79,772	92.59	**95.21**	95.98	95.84	
da	89.46	90.13	89.96	90.55	88,980	94.14	**95.04**	96.13	96.02	96.16
id	89.07	89.34	89.81	90.21	97,531	92.92	93.24	93.41	**93.70**	93.32
zh	80.84	**85.69**	81.84	**85.53**	98,608	90.91	**93.31**	93.36	93.72	
fa	93.50	93.58	93.53	93.71	121,064	96.77	**97.03**	97.20	97.16	97.60
he	90.73	**91.69**	91.93	91.70	135,496	95.65	**96.15**	96.59	96.37	96.62
ro	87.73	**89.18**	88.96	**89.38**	163,262	95.68	**96.72**	97.07	97.09	
en	87.48	**88.45**	88.89	88.89	204,587	93.39	**94.04**	94.90	94.70	95.17
ar	89.01	**90.58**	90.49	90.62	225,853	95.51	**95.72**	96.37	96.24	98.87
hi	87.89	87.77	87.92	88.09	281,057	96.31	96.45	96.64	96.61	96.97
it	91.35	**92.50**	92.45	**93.01**	289,440	97.22	97.47	97.76	97.69	97.90
es	90.54	**91.41**	91.71	91.78	382,436	94.68	94.84	95.08	95.05	95.67
cs	87.97	**90.81**	90.17	**91.29**	1,173,282	96.34	**97.62**	98.18	*97.93*	98.02

Table 5: POS tagging accuracy (UD 1.4 Test). **Bold** (*Italic*) indicates significant improvement (degradation) by McNemar's test, $p < .01$, comparing MIMICK to "No-Char", and "Both" to CHAR→TAG.
* For reference, we copy the reported results of Plank et al. (2016)'s analog to CHAR→TAG. Note that these were obtained on UD 1.2, and without jointly tagging morphosyntactic attributes.

	$N_{train} = 5000$				Full data			
	No-Char	MIMICK	CHAR →TAG	Both	No-Char	MIMICK	CHAR →TAG	Both
kk	—	—	—	—	21.48	20.07	28.47	20.98
ta	80.68	**81.96**	84.26	**85.63**	79.90	**81.93**	84.55	85.01
lv	56.98	**59.86**	64.81	**65.82**	66.16	66.61	76.11	75.44
hu	73.13	**76.30**	73.62	**76.85**	80.04	80.64	86.43	84.12
tr	69.58	**75.21**	75.81	**78.93**	78.31	**83.32**	91.51	90.86
el	86.87	*86.07*	86.40	**87.50**	94.64	**94.96**	96.55	**96.76**
bg	78.26	**81.77**	82.74	**84.93**	91.98	**93.48**	96.12	95.96
sv	82.09	**84.12**	85.26	**88.16**	92.45	**94.20**	96.37	**96.57**
eu	65.29	**66.00**	70.67	*70.27*	82.75	**84.74**	90.58	**91.39**
ru	77.31	**81.84**	79.83	**83.53**	88.80	**91.24**	93.54	93.56
da	80.26	**82.74**	83.59	82.65	92.06	**94.14**	96.05	95.96
zh	63.29	**71.44**	63.50	**74.66**	84.95	85.70	84.86	85.87
fa	84.73	**86.07**	85.94	81.75	95.30	**95.55**	96.90	96.80
he	75.35	68.57	81.06	75.24	90.25	**90.99**	93.35	93.63
ro	84.20	**85.64**	85.61	**87.31**	94.97	**96.10**	97.18	97.14
en	86.71	**87.99**	88.50	**89.61**	95.30	**95.59**	96.40	96.30
ar	84.14	84.17	81.41	*81.11*	94.43	**94.85**	95.50	95.37
hi	83.45	**86.89**	85.64	85.27	96.15	96.21	96.59	**96.67**
it	89.96	**92.07**	91.27	**92.62**	97.32	**97.80**	98.18	98.31
es	88.11	**89.81**	88.58	**89.63**	94.84	**95.44**	96.21	**96.84**
cs	68.66	**72.65**	71.02	**73.61**	91.75	**93.71**	95.29	95.31

Table 6: Micro-F1 for morphosyntactic attributes (UD 1.4 Test). **Bold** (*Italic*) type indicates significant improvement (degradation) by a bootstrapped Z-test, $p < .01$, comparing models as in Table 5. Note that the Kazakh (*kk*) test set has only 78 morphologically tagged tokens.



Part-of-speech tagging (accuracy) Morpho-syntactic attribute tagging (micro-F1)

Figure 2: Results on agglutinative languages (top) and on Slavic languages (bottom). X-axis is number of training tokens, starting at 500. Error bars are the standard deviations over five random training data subsamples.

Test set	Missing embeddings	Full vocabulary		OOV (UD)	
CHAR→TAG		w/o	with	w/o	with
Persian	2.2%	0.03	**0.41**	**0.83**	**0.81**
Hindi	3.8%	**0.59**	0.21	**3.61**	0.36
English	4.5%	**0.83**	0.25	**3.26**	0.49
Spanish	5.2%	0.33	-0.26	1.03	-0.66
Italian	6.6%	**0.84**	0.28	**1.83**	0.21
Danish	7.8%	0.65	**0.99**	**2.41**	**1.72**
Hebrew	9.2%	**1.25**	**0.40**	**3.03**	0.06
Swedish	9.2%	**1.50**	**0.55**	**4.75**	**1.79**
Bulgarian	9.4%	**0.96**	0.12	**1.83**	-0.11
Czech	10.6%	**2.24**	**1.32**	**5.84**	**2.20**
Latvian	11.1%	**2.87**	**1.03**	**7.29**	**2.71**
Hungarian	11.6%	**2.62**	**2.01**	**5.76**	**4.85**
Turkish	14.5%	**1.73**	**1.69**	**3.58**	**2.71**
Tamil*	16.2%	**2.52**	0.35	2.09	1.35
Russian	16.5%	**2.17**	**1.82**	**4.55**	**3.52**
Greek	17.5%	**1.07**	0.34	**3.30**	1.17
Indonesian	19.1%	**0.46**	0.25	**1.19**	0.75
Kazakh*	21.0%	2.01	1.24	**5.34**	**4.20**
Vietnamese	21.9%	0.53	**1.18**	1.07	**5.73**
Romanian	27.1%	**1.49**	**0.47**	**4.22**	**1.24**
Arabic	27.1%	**1.23**	**0.32**	**2.15**	0.22
Basque	35.3%	**2.39**	**1.06**	**5.42**	**1.68**
Chinese	69.9%	**4.19**	**2.57**	**9.52**	**5.24**

Table 7: Absolute gain in POS tagging accuracy from using MIMICK for 10,000-token datasets (all tokens for Tamil and Kazakh). **Bold** denotes statistical significance (McNemar's test,$p < 0.01$).

limited-vocabulary models, without need to access the originating corpus. This method is particularly useful for low-resource languages and tasks with little labeled data available, and in fact is task-agnostic. Our method improves performance over word-based models on annotated sequence-tagging tasks for a large variety of languages across dimensions of family, orthography, and morphology. In addition, we present a Bi-LSTM approach for tagging morphosyntactic attributes at the token level. In this paper, the MIMICK model was trained using characters as input, but future work may consider the use of other subword units, such as morphemes, phonemes, or even bitmap representations of ideographic characters (Costa-jussà et al., 2017).

8 Acknowledgments

We thank Umashanthi Pavalanathan, Sandeep Soni, Roi Reichart, and our anonymous reviewers for their valuable input. We thank Manaal Faruqui and Ryan McDonald for their help in understanding the metrics for morphosyntactic tagging. The project was supported by project HDTRA1-15-1-0019 from the Defense Threat Reduction Agency.

References

Željko Agić, Nikola Ljubešić, and Danijela Merkler. 2013. Lemmatization and morphosyntactic tagging of Croatian and Serbian. In *4th Biennial International Workshop on Balto-Slavic Natural Language Processing (BSNLP 2013)*.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 183–192, Sofia, Bulgaria.

Parminder Bhatia, Robert Guthrie, and Jacob Eisenstein. 2016. Morphological priors for probabilistic neural word embeddings. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Jan A Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Marta R Costa-jussà, David Aldón, and José AR Fonollosa. 2017. Chinese–spanish neural machine translation enhanced with character and word bitmap fonts. *Machine Translation*, pages 1–13.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 4585–4592.

Saso Dzeroski, Tomaz Erjavec, and Jakub Zavrel. 2000. Morphosyntactic tagging of slovene: Evaluating taggers and tagsets. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Denver, CO.

Manaal Faruqui, Ryan McDonald, and Radu Soricut. 2016. Morpho-syntactic lexicon generation using graph-based semi-supervised learning. *Transactions of the Association for Computational Linguistics*, 4:1–16.

Georgi Georgiev, Valentin Zhikov, Petya Osenova, Kiril Simov, and Preslav Nakov. 2012. Feature-rich part-of-speech tagging for morphologically complex languages: Application to Bulgarian. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 492–502, Avignon, France.

Jan Hajič and Barbora Hladká. 1998. Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 483–490.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Miryam de Lhoneux and Joakim Nivre. 2016. Ud treebank sampling for comparative parser evaluation. In *The Sixth Swedish Language Technology Conference (SLTC)*.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 322–332.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. DyNet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Kemal Oflazer and İlker Kuruöz. 1994. Tagging and morphological disambiguation of turkish text. In *Proceedings of the fourth conference on Applied natural language processing*, pages 144–149. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the Association for Computational Linguistics (ACL)*, Berlin.

Cicero D. Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1818–1826.

Natalie Schluter and Željko Agić. 2017. Empirically sampling universal dependencies. In *The NoDaLiDa Workshop on Universal Dependencies (UDW 2017)*.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a tree-bank of modern hebrew text. *Traitement Automatique des Langues*, 42(2):247–380.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the Association for Computational Linguistics (ACL)*, Berlin.