Feature-rich Open-vocabulary Interpretable Neural Representations for All of the World's 7000 Languages

Anonymous ACL submission

Abstract

Modern NLP research is firmly predicated on two assumptions: that very large corpora are available, and that the word, rather than the morpheme, is the primary meaning-bearing unit of language. For the vast majority of the world's languages, these assumptions fail to hold, and as a result existing state-of-the-art neural representations such as BERT fail to meet the needs of thousands of languages. In this paper, we present a novel general-purpose neural representation using Tensor Product Representations that is designed from the beginning to be both linguistically interpretable and fully capable of handling the broad variety found in the world's diverse set of 7000 languages, regardless of corpus size or morphological characteristics. We demonstrate the applicability of our representation through examples drawn from a typologically diverse set of languages whose morphology includes prefixes, suffixes, infixes, circumfixes, templatic morphemes, derivational morphemes, inflectional morphemes, and reduplication.

1 Introduction

Modern NLP research is firmly predicated on two assumptions: that very large corpora are available, and that the word, rather than the morpheme, is the primary meaning-bearing unit of language. English[1] and Standard Mandarin Chinese[2] are the prime examples where both of these conditions hold, and for which existing neural representations such as BERT work very well (Peters et al., 2018; Devlin et al., 2019; Zhang et al., 2019).

1.1 Complex morphology is the norm

The vast majority of NLP research is predicated on the assumption that the word, rather than the morpheme, is the primary meaning-bearing unit of language. This assumption likely stem from the dominance of English as the language of study in NLP (Bender, 2011; Joshi et al., 2020), and the fact that in English, many words do in fact consist of only a single morpheme. Yet for the vast majority of the world's approximately 7000 languages, the average number of morphemes per word is medium or high (see *World Atlas of Language Structures*, including Bickel and Nichols, 2013; Dryer, 2013).

1.2 Unlabelled data is a rare luxury

Somewhere between 100–200 languages (most in the Indo-European language family) have enough unlabelled data (Joshi et al., 2020; Conneau et al., 2020) for BERT embeddings of reasonable quality to be trained using a combination of techniques including unsupervised sub-word segmentation methods, multilingual bootstrapping, and transfer learning. Quality of word embeddings is substantially lower when corpus sizes are insufficiently large; Alabi et al. (2020), for example, construct word embeddings using approximately 10 million tokens for Yorùbá[3] and Twi,[4] and find that the resulting embeddings are substantially poorer in quality those for high-resource languages.

In total, fewer than 300–400 languages have have more than a trivial amount of digitized unlabelled data, thus rendering data-driven NLP approaches including BERT futile for 96% of the world's languages (representing over 1.2 billion people; Vannini and Crosnier, 2012; Joshi et al.,

[1] ISO 639-3: *eng*, an analytic language in the Germanic branch of the Indo-European language family

[2] ISO 639-3: *cmn*, an analytic language in the Sinitic branch of the Sino-Tibetan language family

[3] ISO 639-3: *yor*, an analytic language in the Yoruboid branch of the Niger-Congo language family

[4] ISO 639-3: *twi*, an analytic language in the Tano branch of the Niger-Congo language family

2020), even with aggressive multilingual models, transfer learning, bilingual anchoring, and typologically-aware modelling (Ponti et al., 2019; Michel et al., 2020; Eder et al., 2021; Hedderich et al., 2021).

1.3 Better representations are needed

The current state-of-the-art in neural word representation is insufficient to represent 96% of the world's languages (§1–§2). In this paper, we present a novel general-purpose neural representation (§3) using Tensor Product Representations (TPRs, Smolensky, 1990) that is designed from the beginning to be both linguistically interpretable (§4) and fully capable of handling the broad variety found in the world's diverse set of 7000 languages, regardless of corpus size or morphological characteristics. We demonstrate the applicability of our representation[5] through examples (§4.4) drawn from a typologically diverse set of languages whose morphology includes prefixes, suffixes, infixes, circumfixes, templatic morphemes, derivational morphemes, inflectional morphemes, and reduplication.

2 Existing Word Representations are Insufficient for Most Languages

Computational processing of natural language requires practical digital representations of the words of a language. We survey existing methods for representing words, arguing that while existing word representations work well for high resource analytic languages like English, existing representations are insufficient for effectively representing morphologically complex words in thousands of languages for which large corpora do not exist.

2.1 Representing characters as integers

Oettinger (1954, ch. 2, p. 11), in the very first Ph.D. granted in the field of NLP, defined a word as "any string of letters preceded and followed by a space or a punctuation mark," and stored each word in an electronic dictionary as a sequence of characters, with each character represented digitally as a 5-bit integer. Nearly seventy years later, with relatively minor variations, this definition is still widely used in the NLP research community. Most digital word representations incorporate this technique, storing each character in a word as a multi-bit integer.

[5]Our open source code constructs interpretable word representations from morphologically analyzed examples and trains dense word vectors from the resulting tensors.

2.2 Representing words as feature bundles

During the 1960s through the early 1990s, most NLP systems utilized a knowledge-based paradigm in which words were represented as complex bundles of linguistic features, which were subsequently processed using linguistically-motivated rules (Hutchins, 1986). Finite-state morphological analyzers (Beesley and Karttunen, 2003) can be used to segment words into sequences of component morphemes; such segmentations can include explicit linguistic features such as case, number, and mood in addition to morpheme identity. Another modern example of this type of linguistically feature-rich word representation can be seen in the attribute-value matrices (AVMs) of Head-driven Phrase Structure Grammars (HPSG; Pollard and Sag, 1994). Such linguistically-based feature bundle representations can in principle work with any language, regardless of corpus size or morphological characteristics, but must be constructed by an expert linguist for each language, and do not naturally fit with existing neural techniques.

2.3 Representing words as integers

The development of large digital corpora (primarily in English) and the rise of empirical approaches to NLP in the late 1980s and early 1990s, led to widespread use of statistical language models and translation models (see Church and Mercer, 1993; Manning and Schütze, 1999; Koehn, 2010). When implementing these statistical models, it is often convenient to map each word type to an integer, allowing these integer word representations to directly serve as indices into probability tables (see for example §5 of Brown et al., 1993). A special integer value (often zero) is typically reserved to represent all words not seen during training.

While representing words as integers is efficient in its use of RAM, it suffers from a serious shortcoming first observed by Bull et al. (1955), namely that no semantic, syntactic, or morphological information is encoded in the word representation (for example, *dog* and *dogs* are treated as completely unrelated word types). This problem is seriously exacerbated in languages with rich morphology, as productive derivational and inflectional morphology may result in extremely large numbers of closely-related word types, few of which are likely to appear in corpora. Schwartz et al. (2020), for example, found that in one polysynthetic language, approximately every other word in running text will have never been previously seen.

2.4 Representing subwords as integers

Unsupervised techniques can be used to automatically segment words into sequences of shorter subword tokens generally longer than the character but shorter than the word. These techniques include approaches such as Morfessor (Creutz and Lagus, 2002; Smit et al., 2014) designed to segment words into units approximating morphemes, and compression-based subword segmentation techniques such as BPE (Sennrich et al., 2016; Wu et al., 2016; Kudo and Richardson, 2018). Most neural NLP systems in broad use today utilize integer representations of unsupervised subword tokens for both input and output.

This approach is more successful at representing words in languages with highly productive morphology than the integer word representations described in §2.3. When corpus sizes are small or nonexistent, however, as is the case for most of the world's languages, insufficient training signal exists to reliably train high-quality unsupervised subword segmentation. This problem can be mitigated through the use of a linguistically-based finite-state morphological analyzer (§2.2) for word segmentation instead of unsupervised segmentation methods (Park et al., 2021).

2.5 Representing words as embeddings

Distributed representations (Hinton et al., 1986), also called continuous representations and word embeddings, represent each word as a point embedded in a high-dimensional vector space. When feed-forward or recurrent neural networks are trained as language models with the task of predicting the next word in a sequence, a side effect of the training process is a table of word embeddings which can be indexed by the integer word representations from §2.3. Other techniques for learning context-independent word vector representations for each word type include word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014).

More recent neural techniques such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) can be used to obtain a context-dependent word vector representation for each word token. ELMo uses convolutional techniques to generalize over character sequences within the word in conjunction with deep bidirectional recurrent neural networks, while BERT utilizes unsupervised subword tokenization techniques (§2.4) in conjunction with a transformer architecture (Vaswani et al., 2017).

Learned context-free word embeddings empirically appear to implicitly encode at least some syntactic and semantic information (Mikolov et al., 2013b). Substantial recent work, summarized by Rogers et al. (2020) indicates that contextualized word embeddings learned by BERT are even more successful at implicitly encoding syntactic, semantic, and possibly morphological information. Interpretability of these embeddings is a challenging problem which is far from solved.

While multilingual training, transfer, and anchoring methods have been shown in some cases to somewhat improve the quality of very low-resource word embeddings over monolingually-trained low-resource word embeddings (see, for example, Eder et al., 2021), such methods rely on digitized monolingual and bilingual resources that exist for only a few hundred languages. It remains the case that at present, training high quality word embeddings is dependent on the availability of large corpora (Alabi et al., 2020; Joshi et al., 2020; Wu and Dredze, 2020; Budur et al., 2020; Michel et al., 2020) consisting of tens or hundreds of millions of tokens, which are available for at most a few hundred languages (see §1.2).

2.6 Linguistically-informed word embeddings

No existing word representation is capable of robustly representing words in all of the world's languages regardless of corpus size and morphological characteristics. The existing representation that comes closest to meeting these needs is Linguistically Informed Multi-Task BERT (LIMIT-BERT Zhou et al., 2020b), a semi-supervised approach in which a trained parser (Zhou et al., 2020a) is used to annotate large unlabelled corpora. During LIMIT-BERT training, these silver linguistic annotations (part-of-speech tags, constituency trees, and dependency trees) are used along with the words themselves to train contextualized embeddings on five parsing-related tasks.

Unlike the embeddings learned by LIMIT-BERT, the representations we propose are explicitly interpretable by design, allowing for direct recovery of any linguistic features encoded in our word embeddings. Unlike LIMIT-BERT, our approach can produce high-quality word embeddings in the presence of arbitrarily complex morphology and in the absence of large training corpora.

3 Feature-rich Open-vocabulary Interpretable Representations

We propose a feature-rich open-vocabulary interpretable representation (FOIR) designed to model words from all of the world's languages, even in the absence of a digitized corpus.

3.1 Word Representation Desiderata

Our representation is designed to model words from polysynthetic languages, agglutinative languages, fusional languages, and isolating languages equally well, naturally incorporating any and all linguistic features which may be available from external resources. Our representation is designed to model words in ultra-low-resource settings where corpus sizes are very small or even nonexistent just as well as words in high-resource settings with very large corpora. Our representation is designed to be open-vocabulary, robustly providing word embeddings for novel word types never previously encountered. Finally, our representation is interpretable; all linguistic features encoded in our word embeddings are easily retrievable from the word embeddings.

3.2 Tensor Product Representation

To satisfy the word representation desiderata specified in §3.1, we utilize the Tensor Product Representation (TPR) proposed by Smolensky (1990). The use of TPRs provides a principled way of representing hierarchical symbolic information from external resources such as interlinear glosses or morphological analyzers into vector spaces, such as those used as the input and output domains of neural networks.

Constructing a TPR for a linguistic unit (such as a morpheme or a word) begins by decomposing the symbolic structure of that unit into *roles* and *fillers*. Each role represents a linguistic feature (such as noun case or verb mood), while each filler represents the actual value of that feature (such as associative case or indicative mood). The symbolic structure of a word is then represented as the *bindings* of fillers to roles for all feature-value pairs associated with that unit. Once decomposed, both roles and fillers are embedded into a vector space such that all roles are linearly independent from one another. Let b be a list of ordered pairs (i, j) representing filler i (with embedding vector $\hat{\mathbf{f}}_i$) being bound to role j (with embedding vector $\hat{\mathbf{r}}_j$). The *tensor product representation* $\mathbf{T}$ of the information is then given by

$$\mathbf{T} = \sum_{(i,j)\in b} \hat{\mathbf{f}}_i \otimes \hat{\mathbf{r}}_j \in \mathbb{R}^d \otimes \mathbb{R}^n. \qquad (1)$$

The resulting TPR may itself be used as a filler (for example, the associative case morpheme) and subsequently be bound to another role vector (for example, the noun case of the word). This process results in a TPR that represents the hierarchical compositional structure of a word.

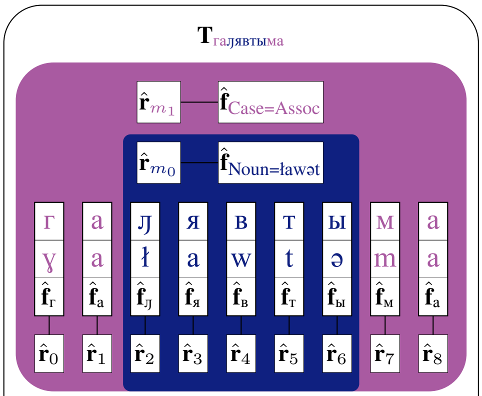
3.3 Robust support for full linguistic diversity

We demonstrate the broad applicability of our feature-rich open-vocabulary interpretable representations (FOIR) using examples drawn from a typologically diverse set of polysynthetic, agglutinative, fusional, and analytic languages. Our examples include prefixes, suffixes, infixes, circumfixes, templatic morphemes, derivational morphemes, inflectional morphemes, and reduplication.

While we expect FOIRs to primarily be constructed using the results of finite-state morphological analyzers and (to a lesser extent) part-of-speech taggers and parsers, in principle FOIRs can be constructed directly from interlinear glosses created by hand by a linguist, even for languages with absolutely no digitized resources or corpora.

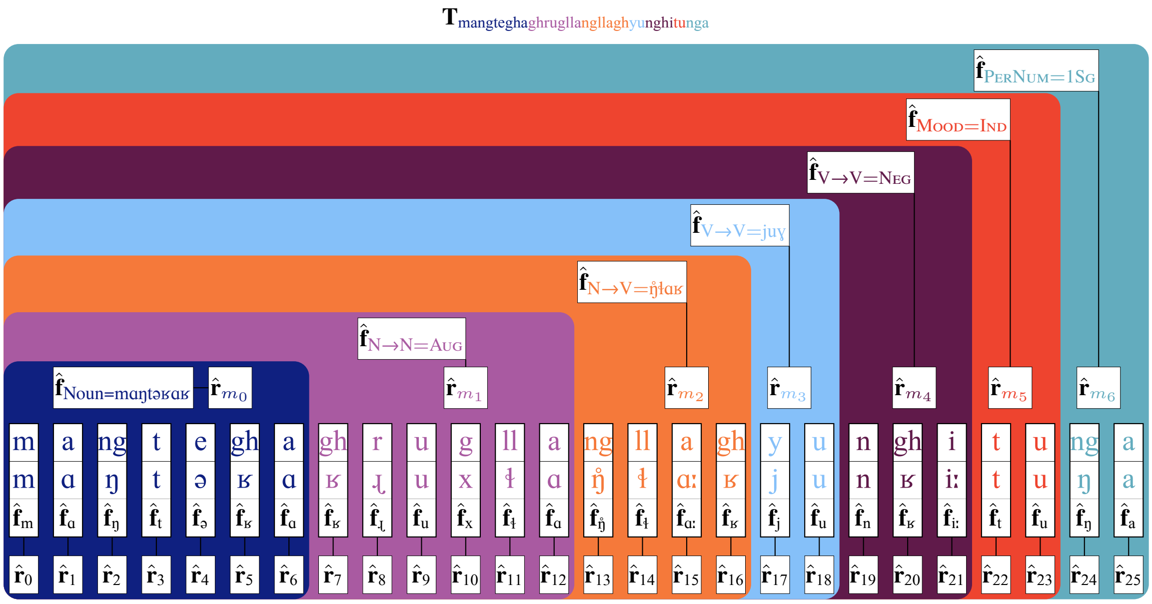
3.3.1 Circumfixes in Chukchi

The Chukchi[6] word гаЈявтыма is composed of a noun root morpheme ɬawət and an inflectional circumfix ɣa…ma. The tensor $\mathbf{T}_{\text{гаЈявтыма}}$ is a TPR that represents this word, ***explicitly including*** all information shown in Example (1):



(1) ɣa- ɬewət -ma
Assoc- head -Assoc
"with the head" (Chukchi)

[6] ISO 639-3: *ckt*, a polysynthetic language in the Chukotkan branch of the Chukotko–Kamchatkan language family



(2)	maŋtəʁaʁ	-ʁɻuxɬaɣ	-ŋ̊ɬaʁ	-juɣ	-nʁitə	-tu	-ŋa
	house	AUG	build	want	NEG	IND	1SG

'I didn't want to make a huge house' (Akuzipik)

The individual characters positions in the word comprise roles $\hat{\mathbf{r}}_0$ through $\hat{\mathbf{r}}_8$, while the characters (and respective phonemes) at those respective positions comprise fillers $\hat{\mathbf{f}}_{\text{г}}$, $\hat{\mathbf{f}}_{\text{а}}$, $\hat{\mathbf{f}}_{\text{ј}}$, $\hat{\mathbf{f}}_{\text{я}}$, $\hat{\mathbf{f}}_{\text{в}}$, $\hat{\mathbf{f}}_{\text{т}}$, $\hat{\mathbf{f}}_{\text{ы}}$, and $\hat{\mathbf{f}}_{\text{м}}$ that encode character and phoneme identity. Roles $\hat{\mathbf{r}}_{m_0}$ and $\hat{\mathbf{r}}_{m_1}$ represent morpheme positions within the word, and are respectively filled by $\hat{\mathbf{f}}_{\text{Noun=ɬawət}}$ (denoting the identity of the root morpheme) and $\hat{\mathbf{f}}_{\text{Case=Assoc}}$ (denoting the identity of the circumfix morpheme marking associative case).

3.3.2 Polysynthesis with derivational and inflectional suffixes in Akuzipik

Productive derivational and inflectional suffixes are pervasive in the polysynthetic languages of the Inuit-Yupik language family. Words with 2-5 derivational morphemes are very common, often representing in a single word what in English would be represented by an entire clause or sentence.

The Akuzipik[7] word mangteghaghrugllanglaghyunghitunga shown in Example (2) can be translated into English as the sentence 'I didn't want to make a huge house' (Jacobson, 2001, pg. 43). The tensor $\mathbf{T}_{\text{mangteghaghrugllanglaghyunghitunga}}$ encodes the hierarchical structure of this word. Each grapheme position within the word is assigned a role ($\hat{\mathbf{r}}_0 \dots \hat{\mathbf{r}}_{25}$). For each of these grapheme position roles, a filler vector encodes the identity of the grapheme and corresponding phoneme at that position in the word ($\hat{\mathbf{f}}_0 \dots \hat{\mathbf{f}}_{25}$). The binding of grapheme position roles to grapheme filler vectors represents the first level of hierarchy in the TPR. The word is composed of 7 morphemes: a noun root maŋtəʁaʁ, four derivational morphemes (-ʁɻuxɬaɣ, -ŋ̊ɬaʁ, -juɣ, -nʁitə) and two inflectional morphemes (-tu and -ŋa). The subsequent levels of the TPR encode the identity, underlying form, surface form, and hierarchical scope of each morpheme. The resulting word representation is compositional and easily interpretable.

By inspecting the resulting tensor, the following structure of the word can be clearly observed:

- The noun root for 'house' maŋtəʁaʁ is modified by the augmentatitive derivational morpheme -ʁɻuxɬaɣ, resulting in an extended noun stem meaning 'big house' spanning grapheme positions 0 through 12.
- The resulting extended noun stem (maŋtəʁaʁɻuxɬaɣ) is verbalized by the derivational morpheme -ŋ̊ɬaʁ, resulting in an extended verb stem meaning 'to build a big house' spanning grapheme positions 0 through 16.
- The resulting extended verb stem (maŋtəʁaʁɻuxɬaŋ̊ɬaʁ) is modified by the derivational morpheme -juɣ, resulting in an extended verb

[7] ISO 639-3: *ess*, a polysynthetic language in the Yupik branch of the Inuit-Yupik-Unangan language family

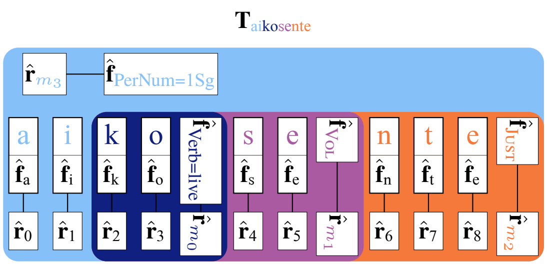
stem meaning ‘to want to build a big house’ spanning grapheme positions 0 through 18.

- The resulting extended verb stem (maŋtəʁa-ʁɹuxɬaŋ̊ɬaʁjuɣ) is modified by the negating derivational morpheme -nʁitə), resulting in an extended verb stem meaning ‘to not want to build a big house’ spanning grapheme positions 0 through 21.

- The resulting extended verb stem (maŋtəʁa-ʁɹuxɬaŋ̊ɬaʁjunʁitə) is marked as being in the indicative mood by the inflectional morpheme -tu and as having a first person singular subject by the inflectional morpheme -ŋa, resulting in the fully inflected word spanning grapheme positions 0 through 25.

3.3.3 Agglutination in Guaraní

In the Guaraní[8] word aikosente shown in Example (3), the verb root ko ‘to live’ is modified in agglutinative manner by two suffixes (-se and -nte) and one inflectional prefix (ai-) which indicates a first person singular subject. Note that unlike the preceding example, which also encoded phoneme identity, in this example character fillers encode only character identity.



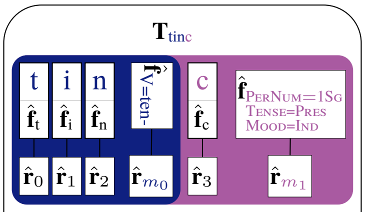
(3) aikosente
ai- ko -se -nte
SG1- live -VOL -JUST
‘I would just like to live’ (Guaraní)

3.3.4 Fusional suffixes in Catalan

Our representation works equally well for simpler examples, such as the Catalan[9] word tinc in Example (4), which is comprised only of only a verb root ten- ‘to have’ and a single inflectional suffix marking person, number, tense, and mood.

[8] ISO 639-3: *gug*, an agglutinative language in the Tupian language family

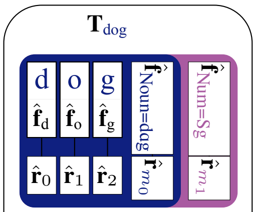
[9] ISO 639-3: *cat*, a fusional language in the Romance branch of the Indo-European language family



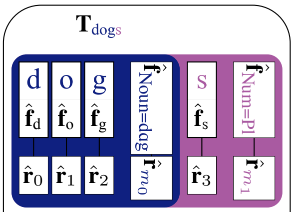
(4) tinc
ten -c
to.have -1SG.PRES.IND
“I have” (Catalan)

3.3.5 Zero inflection & infixation in English

Our representation can encode linguistic features of a word even when those features are not explicitly marked in the surface form of the word. In Example (11), the tensor $\mathbf{T}_{dog}$ explicitly encodes the null singular morpheme -∅ marking number as singular in the word ‘dog,’ just as the morpheme -s marks number as plural in the word ‘dogs in Example (12).’ Unlike existing representations discussed in §2, $\mathbf{T}_{dog}$ and $\mathbf{T}_{dogs}$ are clearly distinguishable as variant inflections of the same root word.

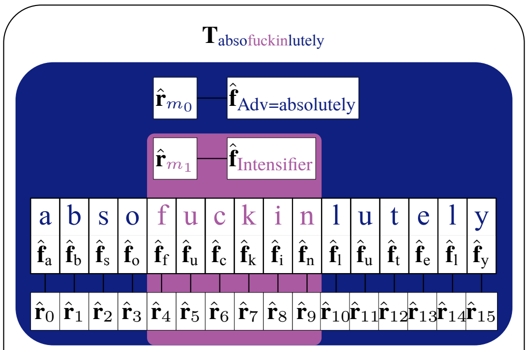


(5) dog -∅
dog -SG
“dog” (English)



(6) dog -s
dog -PL
“dogs” (English)

Linguistic features such as infixes that are attested but relatively rare can also be included with no difficulty. Infixes are morphemes that break a given stem and appear inside it.



(7) abso- fuckin -lutely
abso- INTENSIFIER -lutely
“absofuckinlutely” (English)

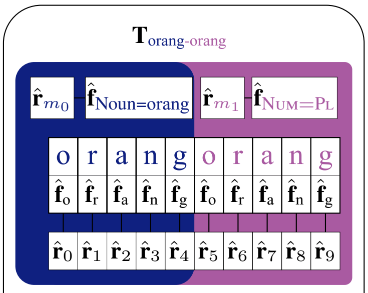
In Seri[10], for example, infixation after the first

[10] ISO 639-3: *sei* a language isolate in north-west Mexico

vowel in the root is used to mark number agreement. In Example (7), we observe an example of expletive infixation in English (McCarthy, 1982) with the infix fuckin serving to intensify the adverb absolutely.

3.3.6 Reduplication in Malaysian

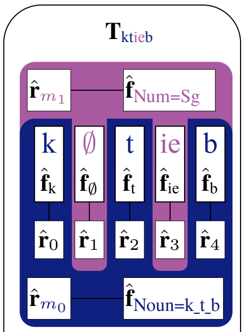
The Malaysian[11] word orang-orang ‘people’, is formed through reduplication of the noun root orang ‘person’. Unlike in previous examples, in which morpheme fillers encoded underlying lexical form in addition to morpheme surface form and identity, in Example (8), the plural morpheme has no underlying lexical form other than the morpheme identity (NUM=PL), as the surface form of the plural morpheme (here, orang) is formed by duplicating the form of the noun to which it attaches.



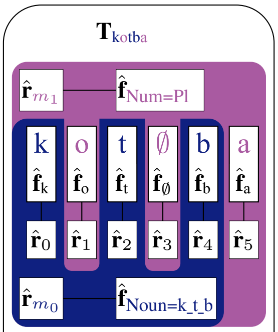
(8) orang -orang
orang -PL
“people” (Malaysian)

3.3.7 Templatic morphology in Maltese

Our representation can easily encode non-concatenative morphology such as that seen in the Maltese[12], words ktieb ‘book’ and kotba ‘books.’



(9) ktieb
k_t_b ∅_ie
book SG
“book” (Maltese)



(10) kotba
k_t_b o_∅_a
book PL
“books” (Maltese)

[11]ISO 639-3: *zsm*, a language in the Malayo-Polynesian branch of the Austronesian language family

[12]ISO 639-3: *mlt*, a templatic language in the Semitic language family

The noun root k_t_b acts as a template whose slots are filled by the vowels in the inflectional singular morpheme ∅_ie (in Example (9)) or plural morpheme o_∅_a (in Example (10)).

4 Embedding and retrieving linguistic information from word vectors

TPRs are useful because they embed arbitrary symbolic structure in a vector space in such a way that simple linear algebra operations may be used to retrieve the form of the symbolic structure, including its compositional structure.

4.1 Learning vectors using an autoencoder

Depending on how much linguistic information is encoded, each of our TPRs may consist of approximately 10^3 to 10^9 floating point values per tensor. Tensors of this size are far too large to be directly usable as neural word representations. To learn lower-dimensional vectors, we make use of an autoencoder. The autoencoder is trained using a dictionary of word or morpheme TPRs. The trained autoencoder can be used to encode a low-dimensional vector from a high-dimensional tensor by running the tensor through the first half of the autoencoder, and can be used to reconstitute the high-dimensional tensor from a vector by running the vector though the latter half of the autoencoder.

4.2 Unbinding

The core operation in retrieving structure from a TPR is called *unbinding*. Exact unbinding requires linear independence of the roles; however, Haley and Smolensky (2020) present an accurate approximate unbinding strategy for even densely packed TPRs. In this work, we use self-addressing unbinding, as it is quick to compute and proved sufficiently accurate for our purposes. Self-addressing unbinding retrieves the filler $\tilde{\mathbf{f}}_i$ for the role $\hat{\mathbf{r}}_i$ by simply computing the inner product between the role vector and the TPR:

$$\tilde{\mathbf{f}}_i = \mathbf{T} \cdot \hat{\mathbf{r}}_i \qquad (2)$$

This unbinding is exact if the role vectors are orthogonal to one another. In our case, since we have a fixed filler vocabulary, we were able to snap our unbindings to the filler with the highest cosine similarity to the unbound vector with sufficient accuracy to render this intrusion irrelevant. Other unbinding strategies involve computing an inverse or pseudoinverse of a matrix of role vectors to perform a change of basis and decrease the intrusion.

4.3 Unbinding loss

In order to effectively train the autoencoder in §4.1, gold standard TPRs must be compared against predicted tensors reconstituted by the autoencoder. However, these tensors are very high dimensional. In initial experiments, we used mean squared error as a loss function, but we found this was unable to converge for auto-encoding sparse TPRs.

To enable effective training of the autoencoder, we therefore define a novel loss function that makes use of the information encoded in the TPR. We define a loss function called *unbinding loss* that examines the unbinding properties of a predicted morpheme tensor to answer the question, "What filler is closest to the unbinding of each role in the TPR?"

Given a predicted tensor, the unbinding loss is computed by recursively unbinding roles until the leaves of the structure are reached – that is, unbind each role until the result of unbinding is a single vector (rather than a higher-order tensor). When this point is reached, we compute the cosine similarity between the result of unbinding and all the fillers in the vocabulary.

This similarity vector can be used to define a probability distribution over possible fillers through the use of a softmax. We take the logarithm of the result of this computation to obtain log-probabilities. We call this distribution P. We then treat each filler (in this case, each character) as a class, and compute the negative log-likelihood loss over this probability distribution.

As we consider tree-structured representations, the number of fillers needing to be checked is exponential with the depth of our representation. This difficulty could be overcome by parallelizing the independent matrix computations for the loss of all the position roles for a given morpheme, trading space for time. For more complex TPRs, a potential avenue would be to exploit the fact that most roles will be empty (and their unbindings thus a matrix of zeros) by replacing the loss computations for unbound roles with mean squared error (which need only push that part of the representation to 0).

See Appendix A for more details on unbinding loss.

4.4 Successfully recovering surface forms from vectors

The Akuzipik data contains 6372 unique morpheme surface forms. Using TPRs constructed from these morphemes, we trained a 3-layer autoencoder with vector sizes of 64, 128, 256, and 512 using unbinding loss (§4.3) as the loss function. We then reconstructed the morpheme surface forms from the trained morpheme vectors. For vector size of 64, the reconstructed morpheme surface form exactly matched the original morpheme surface form for 97.8% of the morphemes. For vector sizes of 128, 256, and 512, the morpheme surface form reconstruction accuracy was 100%.

5 Novel Contributions

In this work, we have defined and implemented[13] a novel general-purpose linguistic representation (§3), taking up the challenge of Church (2011) that "it is better to address the core scientific challenges than to continue to look for easy pickings that are no longer there." Our model is capable of gracefully handling the immense morphological variety and complex hierarchical linguistic structures found across the world's 7000 languages, even in the complete absence of any unlabelled corpora (§1–§2). We have demonstrated our representation using complex examples that include circumfixation (§3.3.1), polysynthesis (§3.3.2), agglutination (§3.3.3), zero inflection (§3.3.5), infixation (§3.3.5), reduplication (§3.3.6), and templatic morphology (§3.3.7). We have defined and implemented[13] a novel loss function that enables successful training of bidirectional mappings between our interpretable sparse tensor representations and equivalent dense vector representations (§4.1–§4.3), and have demonstrated that linguistic information encoded in these vectors can be successfully recovered (§4.4).

References

Jesujoba Alabi, Kwabena Amponsah-Kaakyire, David Adelani, and Cristina España-Bonet. 2020. Massive vs. curated embeddings for low-resourced languages: the case of Yorùbá and Twi. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2754–2762, Marseille, France. European Language Resources Association.

Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Studies in Computational Linguistics. CSLI Publications, Stanford, California.

Emily M. Bender. 2011. On achieving and evaluating language-independence in NLP. *Linguistic Issues in Language Technology*, 6(3):1–26.

[13]URL to be added on acceptance.

Balthasar Bickel and Johanna Nichols. 2013. Inflectional synthesis of the verb. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Emrah Budur, Rıza Özçelik, Tunga Gungor, and Christopher Potts. 2020. Data and Representation for Turkish Natural Language Inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8253–8267, Online. Association for Computational Linguistics.

William E. Bull, Charles Africa, and Daniel Teichroew. 1955. Some problems of the “word”. In William N. Locke and A. Donald Booth, editors, *Machine Translations of Languages*. Greenwood Press, Westport, Connecticut.

Kenneth Church. 2011. A pendulum swung too far. *Linguistic Issues in Language Technology*, 6(3):1–27.

Kenneth W. Church and Robert L. Mercer. 1993. Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics*, 19(1):1–24.

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

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

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

Matthew S. Dryer. 2013. Prefixing vs. suffixing in inflectional morphology. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *World Atlas of Language Structures*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Tobias Eder, Viktor Hangya, and Alexander Fraser. 2021. Anchor-based bilingual word embeddings for low-resource languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 227–232, Online. Association for Computational Linguistics.

Coleman Haley and Paul Smolensky. 2020. Invertible tree embeddings using a cryptographic role embedding scheme. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3671–3683, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

G.E. Hinton, J.L. McClelland, and D.E. Rumelhart. 1986. Distributed representations. In *Parallel distributed processing: Explorations in the microstructure of cognition*, volume 1: Foundations. MIT Press.

W. John Hutchins. 1986. *Machine Translation: Past, Present, Future*. Computers and Their Applications. Ellis Horwood.

Steven A. Jacobson. 2001. *A Practical Grammar of the St. Lawrence Island / Siberian Yupik Eskimo Language, Preliminary Edition*, 2nd edition. Alaska Native Language Center, Fairbanks, Alaska.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, Cambridge, UK.

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.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts.

John J. McCarthy. 1982. Prosodic structure and expletive infixation. *Language*, 58(3):574–590.

Leah Michel, Viktor Hangya, and Alexander Fraser. 2020. Exploring bilingual word embeddings for Hiligaynon, a low-resource language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2573–2580, Marseille, France. European Language Resources Association.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Anthony Oettinger. 1954. *A Study for the Design of an Automatic Dictionary*. Ph.D. thesis, Harvard University, Cambridge, Massachusetts.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology Matters: A Multilingual Language Modeling Analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, 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.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Lane Schwartz, Sylvia Schreiner, and Emily Chen. 2020. Community-focused language documentation in support of language education and revitalization for St. Lawrence Island Yupik. *Études Inuit Studies*, 43(1–2):291–312.

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.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216.

Laurent Vannini and Hervé Le Crosnier, editors. 2012. *Net.lang: Towards the Multilingual Cyberspace*. C&F éditions.

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 *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

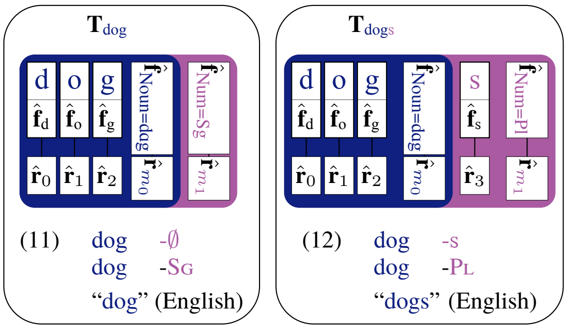
Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Junru Zhou, Zuchao Li, and Hai Zhao. 2020a. Parsing all: Syntax and semantics, dependencies and spans. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4438–4449, Online. Association for Computational Linguistics.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2020b. LIMIT-BERT : Linguistics informed multi-task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online. Association for Computational Linguistics.

A Unbinding loss example



Given a predicted tensor, the first step to computing the unbinding loss is recursively unbinding roles until the leaves of the structure are reached – that is, unbind each role until the result of unbinding is a single vector (rather than a higher-order tensor). When this point is reached, we compute the cosine similarity between the result of unbinding and all the fillers in the vocabulary. For example, assume a depth-4 structure is encoded in a morpheme TPR $\mathbf{T}$, where the fillers are character embeddings, the second level is left-to-right positional roles, the third level is morpheme identity, and the fourth level is left-to-right morpheme position in the word. If we want to see what is bound to the first position of the English *dog* morpheme in $\mathbf{T}$, we would first unbind from $\mathbf{T}$ as follows (assuming self-addressing unbinding):

$$\mathbf{f}_{dog,1} = \mathbf{T} \cdot \hat{\mathbf{r}}_{m0} \cdot \hat{\mathbf{f}}_{Noun=dog} \cdot \hat{\mathbf{r}}_1 \tag{3}$$

We then get the vector of similarities $\hat{\mathbf{s}}_{dog,1}$ between this filler and the each of character embedding vectors in the vocabulary matrix V as follows:

$$\hat{\mathbf{s}}_{dog,1} = \frac{\mathbf{f}_{dog,1} \cdot \mathbf{V}}{||\mathbf{f}_{dog,1}||\mathbf{V}^i\mathbf{V}^i} \tag{4}$$

where $\mathbf{V}^i\mathbf{V}^i$ denotes the column-wise vector norm of the vocabulary matrix (using Einstein summation notation).

This similarity vector can be used to define a probability distribution over possible fillers through the use of a softmax. We take the logarithm of the result of this computation to obtain log-probabilities. We call this distribution P.

$$P = \log\left(\frac{e^{\hat{\mathbf{s}}_{dog,1}}}{\sum e^{\hat{\mathbf{s}}_{dog,1}}}\right) \tag{5}$$

We then treat each filler (in this case, each character) as a class, and compute the negative log-likelihood loss over this probability distribution. The resulting loss for the first character of *dog* being "d" is then

$$loss(\hat{\mathbf{s}}_{dog,1}, d) = -\hat{\mathbf{s}}_{dog,1,d} + \log(\sum_j e^{\hat{\mathbf{s}}_{dog,1,j}}). \tag{6}$$

If the Tensor this loss is computed over is exactly $\mathbf{T}_{\text{dog}}$ or $\mathbf{T}_{\text{dogs}}$, then this loss term would be 0. If we instead considered the loss for the fourth character of the word being "s" in the Num=Pl morpheme, This would be 0 only for $\mathbf{T}_{\text{dogs}}$.