Hierarchical Low-Rank Tensors for Multilingual Transfer Parsing

Hierarchical Low-Rank Tensors for Multilingual Transfer Parsing

Yuan Zhang
CSAIL, MIT
yuanzh@csail.mit.edu

Regina Barzilay
CSAIL, MIT
regina@csail.mit.edu

Abstract

Accurate multilingual transfer parsing typically relies on careful feature engineering. In this paper, we propose a hierarchical tensor-based approach for this task. This approach induces a compact feature representation by combining atomic features. However, unlike traditional tensor models, it enables us to incorporate prior knowledge about desired feature interactions, eliminating invalid feature combinations. To this end, we use a hierarchical structure that uses intermediate embeddings to capture desired feature combinations. Algebraically, this hierarchical tensor is equivalent to the sum of traditional tensors with shared components, and thus can be effectively trained with standard online algorithms. In both unsupervised and semi-supervised transfer scenarios, our hierarchical tensor consistently improves UAS and LAS over state-of-the-art multilingual transfer parsers and the traditional tensor model across 10 different languages.[1]

1 Introduction

The goal of multilingual syntactic transfer is to parse a resource lean target language utilizing annotations available in other languages. Recent approaches have demonstrated that such transfer is possible, even in the absence of parallel data. As a main source of guidance, these methods rely on the commonalities in dependency structures across languages. These commonalities manifest themselves through a broad and diverse set of indicators, ranging from standard arc features used in monolingual parsers to typological properties needed to guide cross-lingual sharing (e.g., verb-subject ordering preference). In fact, careful feature engineering has been shown to play a crucial role in state-of-the-art multilingual transfer parsers (Täckström et al., 2013).

Verb-subject:
{head POS=VERB} ∧ {modifier POS=NOUN} ∧{label=subj} ∧ {direction=LEFT}∧ {82A=SV}
Noun-adjective:
{head POS=NOUN} ∧ {modifier POS=ADJ}∧ {direction=LEFT} ∧ {87A=Adj-Noun}

Table 1: Example verb-subject and noun-adjective typological features. 82A and 87A denote the WALS (Dryer et al., 2005) feature codes for verb-subject and noun-adjective ordering preferences.

Tensor-based models are an appealing alternative to manual feature design. These models automatically induce a compact feature representation by factorizing a tensor constructed from atomic features (e.g., the head POS). No prior knowledge about feature interactions is assumed. As a result, the model considers all possible combinations of atomic features, and addresses the parameter explosion problem via a low-rank assumption.

In the multilingual transfer setting, however, we have some prior knowledge about legitimate feature combinations. Consider for instance a typological feature that encodes verb-subject preferences. As Table 1 shows, it is expressed as a conjunction of five atomic features. Ideally, we would like to treat this composition as a single non-decomposable feature. However, the traditional tensor model decomposes this feature into multiple dimensions, and considers various combinations of these features as well as their individual interactions with other features. Moreover, we want to avoid invalid combinations that con-

[1]The source code is available at https://github.com/yuanzh/TensorTransfer.

join the above feature with unrelated atomic features. For instance, there is no point to constructing features of the form {head POS=ADJ} ∧ {head POS=VERB} ∧ ··· ∧ {82A=SV} as the head *POS* takes a single value. However, the traditional tensor technique still considers these unobserved feature combinations, and assigns them non-zero weights (see Section 7). This inconsistency between prior knowledge and the low-rank assumption results in a sub-optimal parameter estimation.

To address this issue, we introduce a hierarchical tensor model that constrains parameter representation. The model encodes prior knowledge by explicitly excluding undesired feature combinations over the same atomic features. At the bottom level of the hierarchy, the model constructs combinations of atomic features, generating intermediate embeddings that represent the legitimate feature groupings. For instance, these groupings will not combine the verb-subject ordering feature and the POS head feature. At higher levels of the hierarchy, the model combines these embeddings as well as the expert-defined typological features over the same atomic features. The hierarchical tensor is thereby able to capture the interaction between features at various subsets of atomic features. Algebraically, the hierarchical tensor is equivalent to the sum of traditional tensors with shared components. Thus, we can use standard online algorithms for optimizing the low-rank hierarchical tensor.

We evaluate our model on labeled dependency transfer parsing using the newly released multilingual universal dependency treebank (McDonald et al., 2013). We compare our model against the state-of-the-art multilingual transfer dependency parser (Täckström et al., 2013) and the direct transfer model (McDonald et al., 2011). All the parsers utilize the same training resources but with different feature representations. When trained on source languages alone, our model outperforms the baselines for 7 out of 10 languages on both unlabeled attachment score (UAS) and labeled attachment score (LAS). On average, it achieves 1.1% UAS improvement over Täckström et al. (2013)'s model and 4.8% UAS over the direct transfer. We also consider a semi-supervised setting where multilingual data is augmented with 50 annotated sentences in the target language. In this case, our model achieves improvement of 1.7% UAS over Täckström et al. (2013)'s model and 4.5% UAS over the direct transfer.

2 Related Work

Multilingual Parsing The lack of annotated parsing resources for the vast majority of world languages has kindled significant interest in multi-source parsing transfer (Hwa et al., 2005; Durrett et al., 2012; Zeman and Resnik, 2008; Yu et al., 2013b; Cohen et al., 2011; Rasooli and Collins, 2015). Recent research has focused on the non-parallel setting, where transfer is driven by cross-lingual commonalities in syntactic structure (Naseem et al., 2010; Täckström et al., 2013; Berg-Kirkpatrick and Klein, 2010; Cohen and Smith, 2009; Duong et al., 2015).

Our work is closely related to the selective-sharing approaches (Naseem et al., 2012; Täckström et al., 2013). The core of these methods is the assumption that head-modifier attachment preferences are universal across different languages. However, the sharing of arc direction is selective and is based on typological features. While this selective sharing idea was first realized in the generative model (Naseem et al., 2012), higher performance was achieved in a discriminative arc-factored model (Täckström et al., 2013). These gains were obtained by a careful construction of features templates that combine standard dependency parsing features and typological features. In contrast, we propose an automated, tensor-based approach that can effectively capture the interaction between these features, yielding a richer representation for cross-lingual transfer. Moreover, our model handles labeled dependency parsing while previous work only focused on the unlabeled dependency parsing task.

Tensor-based Models Our approach also relates to prior work on tensor-based modeling. Lei et al. (2014) employ three-way tensors to obtain a low-dimensional input representation optimized for parsing performance. Srikumar and Manning (2014) learn a multi-class label embedding tailored for document classification and POS tagging in the tensor framework. Yu and Dredze (2015), Fried et al. (2015) apply low-rank tensor decompositions to learn task-specific word and phrase embeddings. Other applications of tensor framework include low-rank regularization (Primadhanty et al., 2015; Quattoni et al., 2014; Singh et al., 2015) and neural tensor networks (Socher et

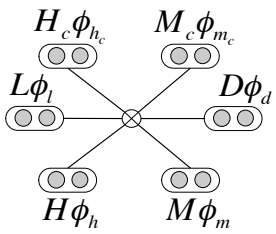

Figure 1: Visual representation for traditional multiway tensor.

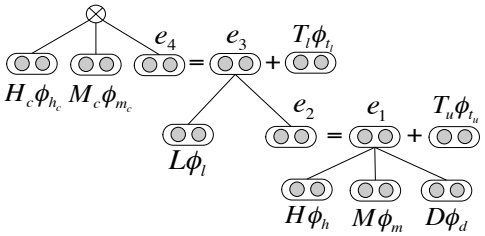

Figure 2: Visual representation for hierarchical tensor, represented as a tree structure. The tensor first captures the low-level interaction ($H\phi_h$, $M\phi_m$ and $D\phi_d$) by an element-wise product, and then combines the intermediate embedding with other components higher in the hierarchy, e.g. e_2 and $L\phi_l$. The equations show that we composite two representations by an element-wise sum.

al., 2013; Yu et al., 2013a). While these methods can automatically combine atomic features into a compact composite representation, they cannot take into account constraints on feature combination. In contrast, our method can capture features at different composition levels, and more generally can incorporate structural constraints based on prior knowledge. As our experiments show, this approach delivers higher transfer accuracy.

3 Hierarchical Low-rank Scoring for Transfer Parsing

3.1 Background

We start by briefly reviewing the traditional three-way tensor scoring function (Lei et al., 2014). The three-way tensor characterizes each arc $h \rightarrow m$ using the tensor-product over three feature vectors: the head vector ($\phi_h \in \mathbb{R}^n$), the modifier vector ($\phi_m \in \mathbb{R}^n$) and the arc vector ($\phi_{h\rightarrow m} \in \mathbb{R}^l$). ϕ_h captures atomic features associated with the head, such as its POS tag and its word form. Similarly, ϕ_m and $\phi_{h\rightarrow m}$ capture atomic features associated with the modifier and the arc respectively. The tensor-product of these three vectors is a rank-1 tensor:

$$\phi_h \otimes \phi_m \otimes \phi_{h\rightarrow m} \in \mathbb{R}^{n\times n\times l}$$

This rank-1 tensor captures all possible combinations of the atomic features in each vector, and therefore significantly expands the feature set. The tensor score is the inner product between a three-way parameter tensor $A \in \mathbb{R}^{n\times n\times l}$ and this rank-1 feature tensor:

$$vec(A) \cdot vec(\phi_h \otimes \phi_m \otimes \phi_{h\rightarrow m})$$

where $vec(\cdot)$ denotes the vector representation of a tensor. This tensor scoring method avoids the parameter explosion and overfitting problem by assuming a low-rank factorization of the parameters A. Specifically, A is decomposed into the sum of r rank-1 components:

$$A = \sum_{i=1}^{r} U(i) \otimes V(i) \otimes W(i)$$

where r is the rank of the tensor, $U, V \in \mathbb{R}^{r\times n}$ and $W \in \mathbb{R}^{r\times l}$ are parameter matrices. $U(i)$ denotes the i-th row of matrix U and similarly for $V(i)$ and $W(i)$. Figure 1 shows the representation of a more general multiway factorization. With this factorization, the model effectively alleviates the feature explosion problem by projecting sparse feature vectors into dense r-dimensional embeddings via U, V and W. Subsequently, the score is computed as follows:

$$S_{tensor}(h \rightarrow m) = \sum_{i=1}^{r} [U\phi_h]_i [V\phi_m]_i [W\phi_{h\rightarrow m}]_i$$

where $[\cdot]_i$ denotes the i-th element of the matrix.

In multilingual transfer, however, we want to incorporate typological features that do not fit in any of the components. For example, if we add the verb-subject ordering preference into $\phi_{h\rightarrow m}$, the tensor will represent the concatenation of this preference with a noun-adjective arc, even though this feature should never trigger.

3.2 Hierarchical Low-rank Tensor

To address this issue, we propose the hierarchical factorization of tensor parameters.[2] The key idea is to generate intermediate embeddings that capture the interaction of the same set of atomic

[2] In this section we focus on delexicalized transfer, and describe the lexicalization process in Section 3.3.

features as other expert-defined features. As Figure 2 shows, this design enables the model to handle expert-defined features over various subsets of the atomic features.

Now, we will illustrate this idea in the context of multilingual parsing. Table 2 summarizes the notations of the feature vectors and the corresponding parameters. Specifically, for each arc $h \rightarrow m$ with label l, we first compute the intermediate feature embedding e_1 that captures the interaction between the head ϕ_h, the modifier ϕ_m and the arc direction and length ϕ_d, by an element-wise product.

$$[e_1]_i = [H\phi_h]_i[M\phi_m]_i[D\phi_d]_i \tag{1}$$

where $[\cdot]_i$ denotes the i-th value of the feature embedding, and H, M and D are the parameter matrices as in Table 2. The embedding e_1 captures the unconstrained interaction over the *head*, the *modifier* and the *arc*. Note that ϕ_{t_u} includes expert-defined typological features that rely on the specific values of the head POS, the modifier POS and the arc direction, such as the example noun-adjective feature in Table 1. Therefore, the embedding $T_u\phi_{t_u}$ captures an expert-defined interaction over the *head*, the *modifier* and the *arc*. Thus e_1 and $T_u\phi_{t_u}$ provide two different representations of the same set of atomic features (e.g. the *head*) and our prior knowledge motivates us to exclude the interaction between them since the low-rank assumption would not apply. Thus, we combine e_1 and $T_u\phi_{t_u}$ as e_2 using an element-wise sum

$$[e_2]_i = [e_1]_i + [T_u\phi_{t_u}]_i \tag{2}$$

and thereby avoid such combinations. As Figure 2 shows, e_2 in turn is used to capture the higher level interaction with arc label features ϕ_l,

$$[e_3]_i = [L\phi_l]_i[e_2]_i \tag{3}$$

Now e_3 captures the interaction between head, modifier, arc direction, length and label. It is over the same set of atomic features as the typological features that depend on arc labels ϕ_{t_l}, such as the example verb-subject ordering feature in Table 1. Therefore, we sum over these embeddings as

$$[e_4]_i = [e_3]_i + [T_l\phi_{t_l}]_i \tag{4}$$

Finally, we capture the interaction between e_4 and context feature embeddings $H_c\phi_{h_c}$ and $M_c\phi_{m_c}$ and compute the tensor score as

Notation	Description
H, ϕ_h M, ϕ_m	Head/modifier POS tag
D, ϕ_d	Arc length and direction
L, ϕ_l	Arc label
T_u, ϕ_{t_u}	Typological features that depend on head/modifier POS but not arc label
T_l, ϕ_{t_l}	Typological features that depend on arc label
H_c, ϕ_{h_c} M_c, ϕ_{m_c}	POS tags of head/modifier neighboring words

Table 2: Notations and descriptions of parameter matrices and feature vectors in our hierarchical tensor model.

$$S_{tensor}(h \xrightarrow{l} m) = \sum_{i=1}^{r}[H_c\phi_{h_c}]_i[M_c\phi_{m_c}][e_4]_i \tag{5}$$

By combining Equation 1 to 5, we observe that our hierarchical tensor score decomposes into three multiway tensor scoring functions.

$$\begin{aligned} S_{tensor}(h \xrightarrow{l} m) &= \sum_{i=1}^{r}[H_c\phi_{h_c}]_i[M_c\phi_{m_c}]_i \\ &\quad \Big\{[T_l\phi_{t_l}]_i + [L\phi_l]_i \\ &\quad \Big([T_u\phi_{t_u}]_i + [H\phi_h]_i[M\phi_m]_i[D\phi_d]_i\Big)\Big\} \\ &= \sum_{i=1}^{r}\Big\{[H_c\phi_{h_c}]_i[M_c\phi_{m_c}]_i[T_l\phi_{t_l}]_i \\ &+[H_c\phi_{h_c}]_i[M_c\phi_{m_c}]_i[L\phi_l]_i[T_u\phi_{t_u}]_i \\ &+[H_c\phi_{h_c}]_i[M_c\phi_{m_c}]_i[L\phi_l]_i[H\phi_h]_i[M\phi_m]_i[D\phi_d]_i\Big\} \end{aligned} \tag{6}$$

This decomposition provides another view of our tensor model. That is, our hierarchical tensor is algebraically equivalent to the sum of three multiway tensors, where H_c, M_c and L are shared.[3] From this perspective, we can see that our tensor model effectively captures the following three sets of combinations over atomic features:

f_1: $\phi_{h_c} \otimes \phi_{m_c} \otimes \phi_{t_l}$
f_2: $\phi_{h_c} \otimes \phi_{m_c} \otimes \phi_l \otimes \phi_{t_u}$
f_3: $\phi_{h_c} \otimes \phi_{m_c} \otimes \phi_l \otimes \phi_h \otimes \phi_m \otimes \phi_d$

[3] We could also associate each multiway tensor with a different weight. In our work, we keep them weighted equally.

The last set of features f_3 captures the interaction across standard atomic features. The other two sets of features f_1 and f_2 focus on combining atomic typological features with atomic label and context features. Consequently, we explicitly assign zero weights for invalid assignments, by excluding the combination of ϕ_{t_u} with ϕ_h and ϕ_m.

3.3 Lexicalization Components

In order to encode lexical information in our tensor-based model, we add two additional components, $H_w\phi_{h_w}$ and $M_w\phi_{m_w}$, for head and modifier lexicalization respectively. We compute the final score as the interaction between the delexicalized feature embedding in Equation 5 and the lexical components. Specifically:

$$[e_5]_i = [H_c\phi_{h_c}]_i[M_c\phi_{m_c}]_i[e_4]_i$$
$$S_{tensor}(h \xrightarrow{l} m) = \sum_{i=1}^{r}[H_w\phi_{h_w}]_i[M_w\phi_{m_w}]_i[e_5]_i \quad (7)$$

where e_5 is the embedding that represents the delexicalized transfer results. We describe the features in ϕ_{h_w} and ϕ_{m_w} in Section 5.

3.4 Combined Scoring

Similar to previous work on low-rank tensor scoring models (Lei et al., 2014; Lei et al., 2015), we combine the traditional scoring and the low-rank tensor scoring. More formally, for a sentence $\mathbf{x}$ and a dependency tree $\mathbf{y}$, our final scoring function has the form

$$S(\mathbf{x},\mathbf{y}) = \gamma \sum_{h \xrightarrow{l} m \in \mathbf{y}} \mathbf{w}\cdot\phi(h \xrightarrow{l} m) + (1-\gamma)\sum_{h \xrightarrow{l} m \in \mathbf{y}} S_{tensor}(h \xrightarrow{l} m) \quad (8)$$

where $\phi(h \xrightarrow{l} m)$ is the traditional features for arc $h \rightarrow m$ with label l and $\mathbf{w}$ is the corresponding parameter vector. $\gamma \in [0,1]$ is the balancing hyper-parameter and we tune the value on the development set. The parameters in our model are $\theta = (\mathbf{w}, H, M, D, L, T_u, T_l, H_c, M_c)$, and our goal is to optimize all parameters given the training set.

4 Learning

In this section, we describe our learning method.[4] Following standard practice, we optimize the parameters $\theta = (\mathbf{w}, H, M, D, L, T_u, T_l, H_c, M_c)$ in a maximum soft-margin framework, using online passive-aggressive (PA) updates (Crammer et al., 2006).

For tensor parameter update, we employ the joint update method originally used by Lei et al. (2015) in the context of four-way tensors. While our tensor has a very high order (8 components for the delexicalized parser and 10 for the lexicalized parser) and is hierarchical, the gradient computation is nevertheless similar to that of traditional tensors. As described in Section 3.2, we can view our hierarchical tensor as the combination of three multiway tensors with parameter sharing. Therefore, we can compute the gradient of each multiway tensor and take the sum accordingly. For example, the gradient of the label component is

$$\partial L = \sum_{h \xrightarrow{l} m \in \mathbf{y}^*} \Big((H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot [(T_u\phi_{t_u}) + (H\phi_h) \odot (M\phi_m) \odot (D\phi_d)]\Big) \otimes \phi_l - \sum_{h \xrightarrow{l} m \in \tilde{\mathbf{y}}} \Big((H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot [(T_u\phi_{t_u}) + (H\phi_h) \odot (M\phi_m) \odot (D\phi_d)]\Big) \otimes \phi_l \quad (9)$$

where $\odot$ is the element-wise product and $+$ denotes the element-wise addition. $\mathbf{y}^*$ and $\tilde{\mathbf{y}}$ are the gold tree and the maximum violated tree respectively. For each sentence $\mathbf{x}$, we find $\tilde{\mathbf{y}}$ via cost-augmented decoding.

Tensor Initialization Given the high tensor order, initialization has a significant impact on the learning quality. We extend the previous power method for high-order tensor initialization (Lei et al., 2015) to the hierarchical structure using the algebraic view as in computing the gradient.

Briefly, the power method incrementally computes the most important rank-1 component for $H(i)$, $M(i)$ etc, for $i = 1 \ldots r$. In each iteration, the algorithm updates each component by taking the multiplication between the tensor T and the rest of the components. When we update the label component l, we do the multiplication for different

[4] Our description focuses on delexicalized transfer, and we can easily extend the method to the lexicalized case.

Feature	Description
82A	Order of Subject and Verb
83A	Order of Object and Verb
85A	Order of Adposition and Noun Phrase
86A	Order of Genitive and Noun
87A	Order of Adjective and Noun

Table 3: Typological features from WALS (Dryer et al., 2005) used to build the feature templates in our work, inspired by Naseem et al. (2012). Unlike previous work (Naseem et al., 2012; Täckström et al., 2013), we use 82A and 83A instead of 81A (order of subject, object and verb) because we can distinguish between subject and object relations based on dependency labels.

multiway tensors and then take the sum.

$$l = \langle T_0, h_c, m_c, -, t_u \rangle + \langle T_1, h_c, m_c, -, h, m, d \rangle$$

where the operator $\langle T_0, h_c, m_c, -, t_u \rangle$ returns a vector in which the i-th element is computed as $\sum_{uvw} T_0(i, u, v, w) h_c(u) m_c(v) t_u(w)$. The algorithm updates other components in a similar fashion until convergence.

5 Features

Linear Scoring Features Our traditional linear scoring features in $\phi(h \xrightarrow{l} m)$ are mainly drawn from previous work (Täckström et al., 2013). Table 3 lists the typological features from "The World Atlas of Language Structure (WALS)" (Dryer et al., 2005) used to build the feature templates in our work. We use 82A and 83A for verb-subject and verb-object order respectively because we can distinguish between these two relations based on dependency labels. Table 4 summarizes the typological feature templates we use. In addition, we expand features with dependency labels to enable labeled dependency parsing.

Tensor Scoring Features For our tensor model, feature vectors listed in Table 2 capture the five types of atomic features as follows:

(a) ϕ_h, ϕ_m: POS tags of the head or the modifier.
(b) ϕ_{h_c}, ϕ_{m_c}: POS tags of the left/right neighboring words.
(c) ϕ_l: dependency labels.
(d) ϕ_d: dependency length conjoined with direction.
(e) ϕ_{t_u}, ϕ_{t_l}: selectively shared typological features, as described in Table 4.

ϕ_{t_l}	$dir \cdot$82A$\cdot\delta(hp$=VERB$\wedge mp$=NOUN$\wedge$subj$\in l)$
	$dir \cdot$82A$\cdot\delta(hp$=VERB$\wedge mp$=PRON$\wedge$subj$\in l)$
	$dir \cdot$83A$\cdot\delta(hp$=VERB$\wedge mp$=NOUN$\wedge$obj$\in l)$
	$dir \cdot$83A$\cdot\delta(hp$=VERB$\wedge mp$=PRON$\wedge$obj$\in l)$
ϕ_{t_u}	$dir \cdot$85A$\cdot\delta(hp$=ADP$\wedge mp$=NOUN$)$
	$dir \cdot$85A$\cdot\delta(hp$=ADP$\wedge mp$=PRON$)$
	$dir \cdot$86A$\cdot\delta(hp$=NOUN$\wedge mp$=NOUN$)$
	$dir \cdot$87A$\cdot\delta(hp$=ADJ$\wedge mp$=NOUN$)$

Table 4: Typological feature templates used in our work. hp/mp are POS tags of the head/modifier. $dir \in \{$LEFT, RIGHT$\}$ denotes the arc direction. 82A-87A denote the WALS typological feature value. $\delta(\cdot)$ is the indicator function. subj $\in l$ denotes that the arc label l indicates a subject relation, and similarly for obj $\in l$.

We further conjoin atomic features (b) and (d) with the family and the typological class of the language, because the arc direction and the word order distribution depends on the typological property of languages (Täckström et al., 2013). We also add a bias term into each feature vector.

Partial Lexicalization We utilize multilingual word embeddings to incorporate partial lexical information in our model. We use the CCA method (Faruqui and Dyer, 2014) to generate multilingual word embeddings. Specifically, we project word vectors in each non-English language to the English embedding space. To reduce the noise from the automatic projection process, we only incorporate lexical information for the top-100 most frequent words in the following closed classes: pronoun, determiner, adposition, conjunction, particle and punctuation mark. Therefore, we call this feature extension partial lexicalization.[5]

We follow previous work (Lei et al., 2014) for adding embedding features. For the linear scoring model, we simply append the head and the modifier word embeddings after the feature vector. For the tensor-based model, we add each entry of the word embedding as a feature value into ϕ_{h_w} and ϕ_{m_w}. In addition, we add indicator features for the English translation of words because this improves performance in preliminary experiments. For example, for the German word und, we add the word and as a feature.

[5] In our preliminary experiments, we observe that our lexicalized model usually outperforms the unlexicalized counterparts by about 2%.

6 Experimental Setup

Dataset We evaluate our model on the newly released multilingual universal dependency treebank v2.0 (McDonald et al., 2013) that consists of 10 languages: English (EN), French (FR), German (DE), Indonesian (ID), Italian (IT), Japanese (JA), Korean (KO), Brazilian-Portuguese (PT), Spanish (ES) and Swedish (SV). This multilingual treebank is annotated with a universal POS tagset and a universal dependency label set. Therefore, this dataset is an excellent benchmark for cross-lingual transfer evaluation. For POS tags, the gold universal annotation used the coarse tagset (Petrov et al., 2011) that consists of 12 tags: noun, verb, adjective, adverb, pronoun, determiner, adposition, numeral, conjunction, particle, punctuation mark, and a catch-all tag X. For dependency labels, the universal annotation developed the Stanford dependencies (De Marneffe and Manning, 2008) into a rich set of 40 labels. This universal annotation enables labeled dependency parsing in cross-lingual transfer.

Evaluation Scenarios We first consider the unsupervised transfer scenario, in which we assume no target language annotations are available. Following the standard setup, for each target language evaluated, we train our model on the concatenation of the training data in all other source languages.

In addition, we consider the semi-supervised transfer scenario, in which we assume 50 sentences in the target language are available with annotation. However, we observe that random sentence selection of the supervised sample results in a big performance variance. Instead, we select sentences that contain patterns that are absent or rare in source language treebanks. To this end, each time we greedily select the sentence that minimizes the KL divergence between the trigram distribution of the target language and the trigram distribution of the training data after adding this sentence. The training data includes both the target and the source languages. The trigrams are based on universal POS tags. Note that our method does not require any dependency annotations. To incorporate the new supervision, we simply add the new sentences into the original training set, weighing their impact by a factor of 10.

Baselines We compare against different variants of our model.

- **Direct**: a direct transfer baseline (McDonald et al., 2011) that uses only delexicalized features in the MSTParser (McDonald et al., 2005).
- **NT-Select**: our model without the tensor component. This baseline corresponds to the prior feature-based transfer method (Täckström et al., 2013) with extensions to labeled parsing, lexicalization and semi-supervised parsing.[6]
- **Multiway**: tensor-based model where typological features are added as an additional component and parameters are factorized in the multiway structure similarly as in Figure 1.
- **Sup50**: our model trained only on the 50 sentences in the target language in the semi-supervised scenario.

In all the experiments we incorporate partial lexicalization for all variants of our model and we focus on labeled dependency parsing.

Supervised Upper Bound As a performance upper bound, we train the RBGParser (Lei et al., 2014), the state-of-the-art tensor-based parser, on the full target language training set. We train the first-order model[7] with default parameter settings, using the current version of the code.[8]

Evaluation Measures Following standard practices, we report unlabeled attachment score (UAS) and labeled attachment score (LAS), excluding punctuation. For all experiments, we report results on the test set and omit the development results because of space.

Experimental Details For all experiments, we use the arc-factored model and use Eisner's algorithm (Eisner, 1996) to infer the projective Viterbi parse. We train our model and the baselines for 10 epochs. We set a strong regularization $C = 0.001$ during learning because cross-lingual transfer contains noise and the models can easily overfit. Other hyper-parameters are set as $\gamma = 0.3$ and $r = 200$ (rank of the tensor). For partial lexicalization, we set the embedding dimension to 50.

7 Results

Table 5 and 7 summarize the results for the unsupervised and the semi-supervised scenarios. Averaged across languages, our model outperforms all

[6] We use this as a re-implementation of Täckström et al. (2013)'s model because their code is not publicly available.

[7] All multilingual transfer models in our work and in Täckström et al. (2013)'s work are first-order. Therefore, we train first-order RBGParser for consistency.

[8] `https://github.com/taolei87/RBGParser`

	Direct		NT-Select		Multiway		Ours	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
EN	65.7	56.7	67.6	55.3	69.8	56.3	**70.5**	**59.8**
FR	77.9	67.4	**79.1**	**68.9**	78.4	68.3	78.9	68.8
DE	62.1	53.1	62.1	53.3	62.1	54.0	**62.5**	**54.1**
ID	46.8	39.3	57.4	37.1	59.5	38.9	**61.0**	**43.5**
IT	77.9	67.9	**79.4**	**69.4**	79.0	69.0	79.3	**69.4**
JA	57.8	16.8	69.2	20.8	69.9	20.4	**71.7**	**21.3**
KO	59.9	34.3	70.4	29.1	70.5	28.1	**70.7**	**30.5**
PT	77.7	71.0	78.5	72.0	78.3	71.9	**78.6**	**72.5**
ES	76.8	65.9	77.2	67.7	77.6	68.0	**78.0**	**68.3**
SV	**75.9**	**64.5**	74.5	62.2	74.8	62.9	75.0	62.5
AVG	67.8	53.7	71.5	53.6	72.0	53.8	**72.6**	**55.1**

Table 5: **Unsupervised:** Unlabeled attachment scores (UAS) and Labeled attachment scores (LAS) of different variants of our model with partial lexicalization in unsupervised scenario. "Direct" and "Multiway" indicate the direct transfer and the multiway variants of our model. "NT-Select" indicates our model without tensor component, corresponding to a re-implementation of previous transfer model (Täckström et al., 2013) with extensions to partial lexicalization and labeled parsing. The last column shows the results by our hierarchical tensor-based model. Boldface numbers indicate the best UAS or LAS.

Feature	Weight
87A$\wedge hp$=NOUN$\wedge mp$=ADJ	2.24×10^{-3}
87A$\wedge hp$=VERB$\wedge mp$=NOUN	8.88×10^{-4}
87A$\wedge hp$=VERB$\wedge mp$=PRON	1.21×10^{-4}
87A$\wedge hp$=NOUN$\wedge mp$=NOUN	9.48×10^{-4}
87A$\wedge hp$=ADP$\wedge mp$=NOUN	3.87×10^{-4}

Table 6: Examples of weights for feature combinations between the typological feature 87A=Adj-Noun and different types of arcs. The first row shows the weight for the valid feature (conjoined with noun→adjective arcs) and the rest show weights for the invalid features (conjoined with other types of arcs).

the baselines in both cases. Moreover, it achieves best UAS and LAS on 7 out of 10 languages. The difference is more pronounced in the semi-supervised case. Below, we summarize our findings when comparing the model with the baselines.

Impact of Hierarchical Tensors We first analyze the impact of using a hierarchical tensor by comparing against the Multiway baseline that implements traditional tensor model. As Table 6 shows, this model learns non-zero weights even for invalid feature combinations.

This disregard to known constraints impacts the resulting performance. In the unsupervised scenario, our hierarchical tensor achieves an average improvement of 0.5% on UAS and 1.3% on LAS. Moreover, our model obtains better UAS on all languages and better LAS on 9 out of 10 languages. This observation shows that the multilingual transfer consistently benefits more from a hierarchical tensor structure. In addition, we observe a similar gain over this baseline in the semi-supervised scenario.

Impact of Tensor Models To evaluate the effectiveness of tensor modeling in multilingual transfer, we compare our model against the NT-Select baseline. In the unsupervised scenario, our tensor model yields a 1.1% gain on UAS and a 1.5% on LAS. In the semi-supervised scenario, the improvement is more pronounced, reaching 1.7% on UAS and 1.9% on LAS. The relative error reduction almost doubles, e.g. 7.1% vs. 3.8% on UAS.

While both our model and NT-Select outperform Direct baseline by a large margin on UAS, we observe that NT-Select achieves a slightly worse LAS than Direct. By adding a tensor component, our model outperforms both baselines on LAS, demonstrating that tensor scoring function is able to capture better labeled features for transfer comparing to Direct and NT-Select baselines.

Transfer Performance in the Context of Supervised Results To assess the contribution of multilingual transfer, we compare against the Sup50 results in which we train our model only on 50 target language sentences. As Table 7 shows, our model improves UAS by 2.3% and LAS by 2.7%. We also provide a performance upper bound

	Semi-supervised Transfer										Supervised Parsing (RBGParser)			
	Direct		Sup50		NT-Select		Multiway		Ours		Partial Lex.		Full Lex.	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
EN	76.8	70.3	79.6	74.2	81.0	75.0	81.5	75.9	**82.5**	**77.2**	88.7	84.5	92.3	90.3
FR	78.8	70.2	76.9	66.8	79.4	71.0	79.0	71.1	**79.6**	**71.8**	83.3	76.5	83.3	76.5
DE	68.4	59.8	71.0	62.4	71.3	62.1	72.1	63.2	**74.2**	**65.6**	82.0	72.8	84.5	78.2
ID	63.7	56.1	78.2	68.9	76.9	68.2	77.8	69.3	**79.1**	**70.4**	85.0	77.1	85.8	79.8
IT	78.9	70.3	77.1	69.3	80.2	72.2	80.8	72.6	**80.9**	**72.6**	85.5	79.8	87.9	84.7
JA	68.2	42.1	**76.6**	61.0	73.0	58.8	75.6	60.9	76.4	**61.3**	79.0	64.0	82.1	70.3
KO	65.3	45.2	70.1	**54.7**	66.5	50.2	67.8	52.8	**70.2**	54.2	74.0	59.1	90.9	86.1
PT	78.6	72.9	76.0	70.0	78.7	73.1	**79.3**	**73.9**	**79.3**	73.5	85.2	80.8	88.5	86.5
ES	77.0	68.5	75.2	66.5	77.0	69.0	77.6	69.5	**78.4**	**70.5**	82.0	75.0	85.8	81.6
SV	77.7	67.2	74.9	64.7	77.6	66.8	77.8	67.5	**78.3**	**67.9**	84.4	75.4	87.3	82.3
AVG	73.4	62.3	75.6	65.8	76.2	66.6	76.9	67.7	**77.9**	**68.5**	82.9	74.5	87.3	83.5

Table 7: **Semi-supervised and Supervised:** UAS and LAS of different variants of our model when 50 annotated sentences in the target language are available. "Sup50" columns show the results of our model when only supervised data in the target language is available. We also include in the last two columns the supervised training results with partial or full lexicalization as the performance upper bound. Other columns have the same meaning as in Table 5. Boldface numbers indicate the best UAS or LAS.

by training RBGParser on the full training set.[9] When trained with partial lexical information as in our model, RBGParser gives 82.9% on UAS and 74.5% on LAS with partial lexical information. By utilizing source language annotations, our model closes the performance gap between training on the 50 sentences and on the full training set by about 30% on both UAS and LAS. We further compare to the performance upper bound with full lexical information (87.3% UAS and 83.5% LAS). In this case, our model still closes the performance gap by 21% on UAS and 15% on LAS.

Time Efficiency of Hierarchical Tensors We observe that our hierarchical structure retains the time efficiency of tensor models. On the English test set, the decoding speed of our hierarchical tensor is close to the multiway counterpart (58.6 vs. 61.2 sentences per second), and is lower than the three-way tensor by a factor of 3.1 (184.4 sentences per second). The time complexity of tensors is linear to the number of low-rank components, and is independent of the factorization structure.

8 Conclusions

In this paper, we introduce a hierarchical tensor based-model which enables us to constrain learned representation based on desired feature interactions. We demonstrate that our model outperforms state-of-the-art multilingual transfer parsers and traditional tensors. These observations, taken together with the fact that hierarchical tensors are efficiently learnable, suggest that the approach can be useful in a broader range of parsing applications; exploring the options is an appealing line of future research.

[9] On average, each language has more than 10,000 training sentences.

Acknowledgments

This research is developed in a collaboration of MIT with the Arabic Language Technologies (ALT) group at Qatar Computing Research Institute (QCRI) within the Interactive sYstems for Answer Search (IYAS) project. The authors acknowledge the support of the U.S. Army Research Office under grant number W911NF-10-1-0533. We thank the MIT NLP group and the EMNLP reviewers for their comments. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors, and do not necessarily reflect the views of the funding organizations.

References

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297. Association for Computational Linguistics.

Shay B Cohen and Noah A Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational*

Linguistics, pages 74–82. Association for Computational Linguistics.

Shay B Cohen, Dipanjan Das, and Noah A Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 50–61. Association for Computational Linguistics.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics.

Matthew S Dryer, David Gil, Bernard Comrie, Hagen Jung, Claudia Schmidt, et al. 2005. The world atlas of language structures.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Cross-lingual transfer for unsupervised dependency parsing without parallel data. *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, page 113.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11. Association for Computational Linguistics.

Jason M Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340–345. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the Annual Conference of the European Chapter of the Association for Computational Linguistics.*, volume 2014.

Daniel Fried, Tamara Polajnar, and Stephen Clark. 2015. Low-rank tensors for verbs in compositional distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325.

Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2014. Low-rank tensors for scoring dependency structures. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1381–1391.

Tao Lei, Yuan Zhang, Regina Barzilay, Lluís Màrquez, and Alessandro Moschitti. 2015. High-order low-rank tensors for semantic role labeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 91–98. Association for Computational Linguistics.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics.

Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 92–97.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244. Association for Computational Linguistics.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 629–637. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Audi Primadhanty, Xavier Carreras, and Ariadna Quattoni. 2015. Low-rank regularization for sparse conjunctive feature spaces: An application to named entity classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Ariadna Quattoni, Borja Balle, Xavier Carreras, and Amir Globerson. 2014. Spectral regularization for max-margin sequence tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1710–1718.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sameer Singh, Tim Rocktaschel, and Sebastian Riedel. 2015. Towards combined matrix and tensor factorization for universal schema relation extraction. In *NAACL Workshop on Vector Space Modeling for NLP*.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 926–934.

Vivek Srikumar and Christopher D Manning. 2014. Learning distributed representations for structured output prediction. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 3266–3274.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242.

Dong Yu, Li Deng, and Frank Seide. 2013a. The deep tensor neural network with applications to large vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(2):388–396.

Mo Yu, Tiejun Zhao, Yalong Bai, Hao Tian, and Dianhai Yu. 2013b. Cross-lingual projections between languages from different families. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 312–317.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 35–42.