Cross-lingual Semantic Role Labelling with the Valpal database knowledge

Chinmay Choudhary
National University of Ireland
Galway
c.choudhary1@nuigalway.ie

Colm O'Riordan
National University of Ireland
Galway
colm.oriordan@nuigalway.ie

Abstract

Cross-lingual Transfer Learning typically involves training a model on a high-resource source language and applying it to a low-resource target language. In this work we introduce a lexical database called **Valency Patterns Leipzig (ValPal)** which provides the argument pattern information about various verb-forms in multiple languages including low-resource languages. We also provide a framework to integrate the ValPal database knowledge into the state-of-the-art LSTM based model for cross-lingual semantic role labelling. Experimental results show that integrating such knowledge resulted in am improvement in performance of the model on all the target languages on which it is evaluated.

1 Introduction

Semantic role labeling (SRL) is the task of identifying various semantic arguments such as Agent, Patient, Instrument, etc. for each of the target verb (predicate) within an input sentence. SRL is useful as an intermediate step in numerous high level NLP tasks, such as information extraction (Christensen et al., 2011; Bastianelli et al., 2013), automatic document categorization (Persson et al., 2009), text-summarization (Khan et al., 2015) question-answering (Shen and Lapata, 2007) etc.
State of the art approaches to SRL such as (Zhou and Xu, 2015; He et al., 2017a,b; Wang et al., 2021) are supervised approaches which require a large annotated dataset to be trained on, thus limiting their utility to only high-resorce languages. This issue of data-sparsity (in low-resource languages) has been effectively addressed with numerous cross-lingual approaches to SRL including *Annotation Projection* approaches (Padó and Lapata, 2009; Kozhevnikov and Titov, 2013; Akbik et al., 2015; Aminian et al., 2019a), *Model Transfer approaches* (McDonald and Nivre, 2013; Swayamdipta et al., 2016; Daza and Frank, 2019; Cai and Lapata, 2020a) and *Machine Translation* approaches (Fei et al., 2020).

In this work, we use the **Valency Patterns Leipzig (ValPal) online database**[1] (Hartmann et al., 2013) which is a multilingual lexical database, originally created by the linguistic research community to study the similarities and differences in verb-patterns for various world languages. Furthermore, we provide a framework to utilise the knowledge available in **Valpal** database to improve the performance of the state-of-the-art cross-lingual approach to SRL task.

2 ValPal Database

Valency Patterns Leipzig (ValPal) is a comprehensive multilingual lexical database which provides semantic and syntactic information about different verb-forms in various languages including many low-resource languages. The ValPal database provides values for the following features for each verb-form:

1. *Valency:* the total number of arguments that a base verb-form can take.

2. *Argument-pattern*: the type and order of arguments taken by a base verb-form in its most common usage.

3. *Alterations:* the alternate *argument-patterns* that can be taken by either the base verb-form or any of its morphological variant.

Table 1 depicts the information about three lexical units namely ***cook***, ***kochen*** and ***cuocere*** as provided in the ValPal database. Please note that Table 1 lists only a few of all the alterations provided for these verb-forms in ValPal database due to space constraints. Lexical units ***cook***, ***kochen*** and ***cuocere*** are *English*, *German* and *Italian* words representing base verb-form for verb activity COOKING.

[1] http://ValPal.info/

Proceedings of Deep Learning Inside Out (DeeLIO 2022):
The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 1 - 10

2.1 Coding of Argument-patterns

In ValPal database each argument-pattern (including alteration) is coded with a unique coding-frame. For example, in Table 1, the argument-pattern of English base verb-form ***cook***, is coded as follows

$$1 - nom > V.subj[1] > 2 - acc$$

The code indicates that the base verb-form ***cook*** takes 2 arguments in its most common usage (valency of 2). The first argument is *cooker* (indicated as *1-nom*) and the second one is *Cooked-food* (indicated as *2-acc*). $V.subj[1]$ indicates the verb with the first argument as its agent. The order of arguments are **cooker–V–cooked_food** (eg: She is cooking the fish.).

Verb-form ***cook*** also has an *alteration* called **Causative-Inchoative** with the derived argument-pattern as follows.

$$2 - acc > V.subj[1]$$

This argument pattern indicates that verb-form can also have order of arguments as **cooked_food–V** with *Agent* argument missing from the sentence (eg: The fish is cooking).

2.2 Coding-sets

ValPal provides a unique coding-set for each language. The codes in these coding sets indicate various argument-types including modifier argument-types. For example, codes NP-Nom, NP-acc and LOC-NP indicate the AGENT (Arg0), PATIENT (Arg1) and modifier LOCATION (ArgM-LOC) arguments respectively in the coding-sets of all languages. The codes with+NP and mit+NP-dat indicate INSTRUMENT argument in English and German coding-sets. Similarly, codes UTT-NP indicate the argument TEMPORAL in most coding-sets. In these codes, the *NP* indicates the index of valency occupied the respective argument within the argument pattern (eg: code $2 - acc$ in argument pattern $2 - acc > V.subj[1]$ indicates argument-type PATIENT with the valency-index of 2).

2.3 Alteration Types

As already explained, the ValPal database also provides a list of alternate argument-patterns (called alterations) for each verb-form. Some of these alterations are *morpho-independent* as they can be taken by the respective base-verb in any morphological form, whereas others are *morpho-dependent* as they can be taken by the respective verb only in a specific morphological form.

For example, both the *Reflexive-Passive* and *Impersonal Passive* alterations of the italian base verb-form *cuocere*, outlined in Table 1 are morpho-dependent alterations as these alterations are observed only when the verb-form possesses morpheme si.

The ValPal database is originally created by the linguistic research community, typically to study the similarities and differences in verb-patterns for various world languages. However this knowledge can also be used by NLP research community for building the models for data-sparse languages.

2.4 FrameNet to aid ValPal

One shortcoming of the Valpal database is that its vocabulary is limited for many languages. If we encounter a verb in the training-set that is missing in ValPal, we utilised the *FrameNet* database to extract the desired *argument-pattern* and *alterations* of it from ValPal itself.

To extract this knowledge about the missing verb, firstly we extracted the frame of the missing verb from the respective *FrameNet* database. Subsequently we extracted a replacement-verb that belongs to the same frame (as that of the missing verb) and is available in ValPal database. Finally, we assigned the argument-pattern and alterations of this replacement-verb to the missing verb. For example, the verb **barbecue** is missing from ValPal database. Yet, the verb **barbecue** belongs to frame **COOKING-45.1** in *English FrameNet* (Barkley). Another verb-form called **cook** belong to the same frame (**COOKING-45.1**) and is available in ValPal database. Thus we use argument-patters provided in ValPal for verb-form **cook** as the argument-patterns for **barbecue**.

3 FOL rules from ValPal

To inject the entire ValPal database knowledge about any low-resource target-language l in a Cross-lingual Neural Network model, we represented this knowledge as a set of First-order-logic (FOL) rules F_l. The process of generating this set of FOL rules involves two steps namely *Translating ValPal Argument-patterns to Propbank label orders* and *Writing Propbank-label order as FOL rule* described in Sections 3.1 and 3.2.

In ValPal database, the argument-pattern for verb-

Verb-form	**Lang**	**Argument-pattern**	**Alterations** (Alteration-name:Arg-pattern (example))
cook	English	$1-nom > V.subj[1] > 2-acc$	***Understood Omitted Object***:$1-nom > V.subj[1] > 2-acc$ (She walked in while I was cooking.) ***Causative-Inchoative*** : $2-acc > V.subj[1]$ (The soup is still cooking.)
kochen	German	$1-nom > V.subj[1] > 2-acc$	***Benefactive Alternation***:$1-nom > V' > subj[1] > 3-dat > 2-acc$ (Ich koche meiner Mutter eine Suppe.) ***be-Alternation***:$1-nom > beV'.subj[1] > 4-acc > mit+2-dat$ (Die Großmutter bekocht die Kranke mit Suppe.) ***Ambitransitive Alternation***:$2-nom > V'.subj[2]$ (Das Wasser kocht.)
cuocere	Italian	$1 > V.subj[1] > 2$	***Reflexive-Passive***:$2 > siV'.subj[2] > daParteDi+1$ (La carne si cuoce con attenzione.) ***Impersonal Passive***:$siPassV' > da+1$ (Quando si è (stati) cotti dal sole si diventa di color rosso intenso.)

Table 1: Sample verb-form knowledge in Valpal database

form ***tie*** is outlined as equation 1 (as Q). We use this as an example to demonstrate the process of converting an argument-pattern to a FOL rule.

$$Q = 1-nom > V.subj[1] > 2-acc > LOC-3(> with+4) \quad (1)$$

3.1 Translate argument-patters to Propbank Order

In this step, we translate all the Valpal's argument-patterns (including alterations) for all lexical verb-forms in the target-language l, to the Propbank Orders. The entire process of translating a ValPal argument-pattern P of any language l into a *Propbank Label-order* involves two simple text-processing sub-steps described as sections 3.1.1 and 3.2.

3.1.1 Replace modifier argument-types

As already explained in section 2.2, the Valpal database provides a unique coding-set for each language. In this subset, we examined the entire coding-set for language l to identify the codes that refer to a modifier argument-type (eg: LOC-NP and UTT-NP etc. in English coding-set for LOCATION and TEMPORAL modifier-arguments), and created a mapping table that maps these modifier-argument codes to the corresponding Propbank annotations (eg: LOC-NP mapped to ARGM-LOC; UTT-NP mapped to ARGM-TMP etc.). The coding-set of any language in the ValPal database is small thus making it feasible to manually create such mapping table.
Subsequently, we used this mapping table to replace all modifier argument-patterns (if any) in the argument-pattern P being translated, with corresponding Propbank label.
After replacing the modifier argument-types we reduce the valency-index of all the arguments following the replaced modifier argument, in the argument-pattern being translated, by one.

$$Q = 1-nom > V.subj[1] > 2-acc > ARGM-LOC(> with+3) \quad (2)$$

For example, the argument-pattern outlined in equation 1 comprises only one modifier argument-type namely LOC3. We replaced this with the corresponding Propbank label namely ARGM-LOC and reduced the valency-index of all argument-types following this replaced argument-pattern by 1 (thus $(with + 4)$ is re-written as $(with + 3)$). Hence the argument-pattern in Equation 1 would be re-written as equation 2.

3.1.2 Rewrite all non-modifier argument types

After replacing all modifier argument-types in the argument-patterns by the process described in section 3.1.1, we simply replace all left over arguments in the ValPal argument-pattern P by string as 'ARGx' where x is $valencyIndex - 1$. Hence argument $1 - nom$, $2 - acc$ and $with + 3$ (with valency Indexes as 1, 2, 3 respectively) in equation 2 would be replaced by *Arg0*, *Arg1* and *Arg2* respectively.
Finally, we replaced $Vsubj[NP]$ with V and removed all bracket symbols. Hence argument-pattern outlined as equation 2 would be translated as equation 3.

$$Q = ARG0 > V > ARG1 > ARG-LOC > ARG2 \quad (3)$$

3.2 Write Propbank Label order as FOL rule

Once having represented all argument-patterns (including alterations) for all lexical verb-forms of language *l* as allowed Propbank Label-orders, we rewrite each verb-form and Propbank Label-order pair as a FOL rule. For example the pair of verb-form ***tie*** and its corresponding allowed Propbank Label-order outlined as equation 4, is represented by the FOL rule indicated as equation 5.

$$f = baseForm(V, tie) \lor pattern(Y, Q) \quad (4)$$

Here Q is the Propbank label-order outlined in equation 3, and Y is the sequence of Propbank tag-sequence predicted by a neural-network model for any input token-seq. The logic-constraint in equation 5 would be true if the verb for which the arguments are being predicted is a variant of base verb-form *tie* and the predicted SRL tag sequence Y satisfies the label order Q.
While checking whether a predicted SRL tag sequence follows a specific order, we ignore the 'O' annotations ('O' indicates semantic role label 'NULL' in the Propbank Annotation scheme). For example the SRL tag sequences ***ARG0, ARG0, O, O, V, ARG1, ARG-LOC, O, ARG2*** follows the argument-pattern.
To check if the verb for which the arguments are being predicted is a morphological variant of the specific base verb-form, we perform stemming of both base verb-form and the token from the sentence which is tagged 'V' by the model. If the stem strings are equal we consider the verb token to be a variant of base verb-form.
If an argument-pattern (represented as Propbank label-order) is for a morpho-dependent alteration, then the morphological constraint is also added to the FOL rule representing the argument-pattern. For example, in table 1 the argument-pattern *Reflexive-Passive* is a morpho-dependent alteration. This argument-pattern is represented as FOL defined by equation 6.

$$f = baseForm(V, cuocere) \lor morphoForm(V, si) \lor pattern(Y, \hat{Q}) \quad (5)$$

Here $\hat{Q}$ represents the corresponding label-sequence for Argument-pattern. The rule $morphoForm(V, si)$ constraints the verb V to have morphene *si* for the rule to be true.
Hence we obtain a set of FOL rules F_l representing the entire Valpal database knowledge about language l (with each verb-form and argument-patterns pair provided in the Valpal database for the language l as a single FOL-rule $f \in F_l$). These FOL rules are used during the fine-tuning of a cross-lingual neural-network model for SRL in target-language l. During fine-tuning, the model is always rewarded if it predicts an SRL tag-seq Y which satisfies atleast one of the FOL rule $f \in F_l$, and penalised otherwise. Section 4.3 will explain the fine-tuning process in more detail.

4 Model

4.1 Base Approach

We utilized the state-of-the-art approach to Cross-lingual SRL in low-resource languages, proposed by (Cai and Lapata, 2020b) as our *Base Approach*. The approach comprises two key components namely *Semantic Role Labeler* and *Semantic Role Compressor*. The Semantic Role Labeler is a simple Bi-LSTM model with Biaffine Role Scorer (Dozat and Manning, 2016). Given input sentence $X = x_1...x_T$ of length T, the model accepts pre-trained multilingual contextualized word-embedding e_{x_i} and predicate indicator embedding p_{x_i} for all $x_i \in X$ as input. For each

word $x_i \in X$, the topmost biaffine layer computes the scores of all semantic roles to be assigned to x_i as $s_i \in R^{|n_r|}$ where n_r is the size of semantic role set. Hence the probability values of all SRL labels to be assigned to word x_i can be computed by applying the softmax function over s_i.
Subsequently, the Semantic Role Compressor is another Bi-LSTM model which compresses the useful information about arguments, predicates and their roles from the outputs of the Semantic Role Labeller (e.g., by automatically filtering unrelated or conflicting information) in a matrix $R \in R^{n_r * d_r}$ where d_r denotes the length of hidden representation for each semantic role.
The approach assumes the availability of a fully annotated source language corpus and parallel corpus of source-target sentences for training. Each model-training step involves two independent sequential sub-steps namely the *the supervised training* and *the cross-lingual training*.
In the source-language training sub-step, a batch is randomly selected from the annotated source-language corpus, to train both *Semantic Role Labeler* and *Semantic Role Compressor* simultaneously by minimizing the total loss computed by equation 3.

$$L_{total} = L_{CE} + L_{KL} \tag{6}$$

Here L_{CE} is the Cross-entropy loss between true labels and labels predicted by the Labeler whereas L_{KL} is the KL Divergence loss (Kullback and Leibler, 1951) between distributions predicted by the Compressor and the Labeler. After the *supervised training* sub-step, a batch from the parallel source-target data to perform the *cross-lingual training* sub-step. We refer to the original work (Cai and Lapata, 2020b) for the details of the *cross-lingual training* sub-step and the inference.

4.2 Training with Valpal knowledge

In this work we modified the training process described in section 4.1 to include the Valpal knowledge into the model parameters. Each training step in our proposed training step involves four independent sequential sub-steps.

Firstly, in the *Labeler pre-training* sub-step, we randomly sample a batch from the annotated source-language corpus and the Semantic Role Labeler is trained on it by minimizing the cross-entropy loss (L_{CE}) between true and predicted roles. Secondly, in the *Labeler fine-tuning*, the Valpal knowledge is injected in the parameters of the Semantic Role Labeler by the process described in section 4.3. Thirdly, in the *Compressor training* sub-step the *Semantic Role Compressor* is trained on the sampled source-language batch by minimizing the KL Divergence loss (L_{KL}) between distributions predicted by the Compressor and the fine-tuned Labeler (Labeler parameters are fixed in this sub-step). Finally we perform the *cross-lingual training* sub-step which is identical to as performed by the original authors (section 4.1)

4.3 Labeler fine-tuning with ValPal

This section describes the framework adopted by us to induce the target-language specific ValPal database knowledge expressed as a set of FOL rules F_l, into the pre-trained *Semantic Role Labeler*. Our framework is inspired by the *Deep Probabilistic Logic* (DPL) framework proposed by (Wang and Poon, 2018). The framework assumes the availability of only an unlabelled target-language corpus. Hence, for the *Labeler fine-tuning* sub-step, we randomly sample a batch from the already available parallel source-target data and utilised only the target language part of it.
Let $X = x_1.....x_T$ be an input sentence and $Y = y_1.....y_T$ be any SRL-tag sequence. Further let Ψ be the pre-trained Bi-LSTM based *Semantic Role Labeler*, such that $\Psi(X, Y)$ denotes the conditional probability $P(Y|X)$ as outputted by the final softmax layer of Ψ.
The fine-tuning of this pre-trained Ψ to specific target-language l requires an unlabelled target-language training corpus. Given such unlabelled target-language-corpus X_{targ}, for each $X \in X_{targ}$ we input sentence X into the pre-trained Ψ to compute the most probable SRL-tag sequence Y as $Y = argmax_{\hat{Y}}(\Psi(x, \hat{Y}))$. Subsequently we input both the sentence X and it's predicted most-probable SRL tag-seq Y in all the FOL rules in F_l to compute their value (as 0.0 or 1.0). DPL framework defines the conditional probability distribution $P(F_l, Y|X)$ as equation 2.

$$P(F_l, Y|X) = \prod_{f \in F_l} \frac{exp(w.f(X,Y)).\Psi(X,Y)}{exp(w)} \tag{7}$$

The framework assumes the Knowledge-constraints to be log-linear thus defines each knowledge-constraint as $exp(w.f(X, Y))$ where $f \in F_l$ is the FOL rule representing the respective knowledge-constraint. Here w is the pre-decided reward-weight assigned to all constraints. Hence the predicted output-sequence Y would be rewarded (as its likelihood would increase by a factor of $exp(w)$) if it follows one of the defined argument-patterns in ValPal database for the respective verb for which the arguments are being predicted ($f(X, Y) = 1.0$). However no penalty is awarded for not following the correct Argument-pattern.

4.3.1 Learning

The ideal way to optimize the weights (fine-tune) of the model Ψ is by minimizing $P(F_l|X)$ and updating the parameters through back-propagation. We can compute $P(F_l|X)$ by summing over all possible SRL-tag sequences as $P(F_l|X) = \Sigma_Y P(F_l, Y|X)$. However computing $P(F_l, Y|X)$ by equation 4 with all possible output-sequences, and subsequently back-propagating through it, for each training example is computationally very inexpensive. Thus DPL framework also provides a more efficient EM-based approach (Moon, 1996) to the parameter fine-tuning which is adopted by us.

The full process of learning the parameters of Ψ (initialized with parameters pre-trained on source language) is outlined as Algorithm 1. For each

---

Algorithm 1 Fine-tuning of Semantic Role Labeller

Require: Target Language corpus X_{targ}; set of FOL rules F_l representing the entire Valpal database knowledge; Pre-trained LSTM based Semantic Role Labeller Ψ; Number of Epochs N;

```
repeat
    for each X ∈ X_targ do
                                        ▷ E-Step
        Y ← argmax_Ŷ(Ψ(X, Ŷ))
        q(Y) ← P(F_l, Y|X)        ▷ by equation 7
                                        ▷ M-Step
        Ψ ← argmin_Ψ̂(D_KL(q(Y)||Ψ̂(X, Y)))
    end for
until convergence
```

---

training-example $X \in X_{targ}$, the Algorithm 1 implements three steps. In the first-step, it predicts the most probable SRL-tag sequence Y for the given training-example X as $Y = argmax_{\hat{Y}}(\Psi(x, \hat{Y}))$ with current parameter values for Ψ.

In the E-step, $q(Y) = P(F_l, Y|X)$ is computed by applying equation 4 with current parameters of Ψ. Finally in the M-step it keeps $q(Y)$ as fixed and updates parameters of Ψ by minimizing the KL-divergence (Kullback and Leibler, 1951) loss between $q(Y)$ and the probability of Y from $\Psi(X, Y)$ (i.e. $P(Y|X)$).

In other words, in each epoch step, the model first computes the joint likelihood of F_l and Y i.e $P(F_l, Y|X)$ with current model parameters, and subsequently it updates the parameters to predict likelihood of Y i.e., to be as close to $P(F_l, Y|X)$ as possible.

5 Experiments

This section described the experiments performed by us to evaluate the proposed model.

5.1 Dataset

We experimented with four languages namely *English* (en), *German* (de), *Chinese* (zh) and *Italian* (it) as these languages are covered in both the ValPal database as well as in the CoNLL 2009 Shared task (Hajic et al., 2009) dataset. The *Semantic Role Labeller* requires a fully-annotated training dataset in the high-resource source-language. We utilized the Universal Proposition Banks provided at `https://github.com/System-T/UniversalPropositions` provided for CoNLL 2009 Shared task, for training of the *Semantic Role Labeller* and the evaluation of various systems. On the other hand, the *Semantic Role Compressor* component requires sentence-paired parallel corpora in source and target languages. We used the Europarl parallel text-corpus (Koehn et al., 2005), and the large-scale EN-ZH parallel corpus (Xu, 2019) to train the *Semantic Role Compressor*, as used by (Cai and Lapata, 2020b). We used the target-language part of the same parallel-corpora independently for the Valpal knowledge induction, as the Valpal database knowledge induction simply requires unlabelled text-corpus in the target-language.

5.2 Model-configurations

We computed the language-independent BERT-Embeddings to be fed into the networks using pre-trained Multilingual BERT (mBERT) (Wu and Dredze, 2019) model. Given a sentence S, we tokenised the whole sentence using the WordPiece tokeniser (Wu et al., 2016). Subsequently we fed

Dropout prob.	0.01
Bach-size	32
Epochs	150
embeddings size	768
predicate indicator embed size	16
Bi-LSTM hidden states size	400
BiLSTM depth	3
hidden biaffine scorer size	300
Bi-LSTM hidden states size	256
BiLSTM depth	2
compressed role rep size	30
hidden biaffine scorer size	30

Table 2: Hyper-parameter settings for input and training (first block), semantic role labeler (second block) and semantic role compressor (third block). Semantic role labeler and Semantic role compressor are same as (Cai and Lapata, 2020b)

this token-sequence into pre-trained mBERT provided by (Turc et al., 2019). Embedding of any word $w \in S$ i.e. e_w is computed by taking average of mBERT outputs of all Wordpiece tokens corresponding to word w. Subsequently these word-embeddings are frozen during the training of the networks. Table 2 outlines the hyper-parameters used during training.

5.3 Baselines

We compared the performance of our proposed model against the base-model (4.1) as well as numerous other state-of-the-art baselines. These baselines include two annotation projection based models namely *Bootstrap* (Aminian et al., 2017) and *CModel* (Aminian et al., 2019b), as well as two strong mixture-of-experts models namely *MOE* (Guo et al., 2018) which focus on combining language specific features automatically as well as *MAN-MOE* (Chen et al., 2018) which learns language-invariant features with the multinomial adversarial network as a shared feature extractor. We also compared with PGN (Fei et al., 2020) which is the state-of-the-art translation-based model which translates the source annotated corpus into the target language, performs annotation projection, and subsequently trains the model on both source and the translated corpus. We utilised the source-code provided by the authors of each of these baselines to train and test them.

Algorithm 2 Full Training process. Here, the function *FineTune* represents the process outlined as algorithm 1 and function *CrossTrain* represents the cross-lingual training procedure adopted by (Cai and Lapata, 2020b). L_{CE} is cross-entropy loss and L_{KL} is KL divergence loss

Require: Annotated Source language corpus $\{X_{Tagged}, Y_{Tagged}\}$; Parallel Source-target Corpus $\{X^S_{Parallel}, X^T_{Parallel}\}$; set of FOL rules representing entire Valpal db knowledge of target language F_l; batch-size b; Number of Epochs E
Initialize:
  Semantic Role Labeler Ψ; Semantic Role Compressor Φ
$steps \leftarrow |X_{Tg}|/b$
for $epoch \leftarrow 1$ to E **do**
  for $step \leftarrow 1$ to $steps$ **do**
    $X, Y \leftarrow$ Sample($\{X_{Tg}, Y_{Tg}\}$,b)
    $X^S, X^T \leftarrow$ Sample($\{X^S_{Pr}, X^T_{Pr}\}$,b)
      ▷ **Labeler pre-training**
    $\Psi \leftarrow argmin_{\hat{\Psi}}(D_{CE}(Y||\hat{\Psi}(X)))$
      ▷ **Labeler Fine-tuning**
    $\Psi \leftarrow FineTune(X^T, F_L, \Psi, b)$
      ▷ **Compressor training**
    $\Phi \leftarrow argmin_{\hat{\Phi}}(D_{KL}(\Psi(X)||\hat{\Phi}(X)))$
      ▷ **Cross-lingual training**
    $\Phi, \Psi \leftarrow CrossTrain(X^S, X^T, \Psi, \Psi)$
  end for
end for

6 Results

6.1 Monolingual training

In the first set of experiments we trained the models on a single source language English and tested these on the target languages *zh*, *it* and *de*. In these settings, we trained the models on English UPB train-dataset and tested them on the UPB test-sets of the target-languages. Table 3 shows the labeled F-scores achieved on each of these target-languages. In table 4, the *Base-wo-Compressor* refers to the base model without the SRL compressor, whereas *Base-full* refers to the full base model.

Results in Table 3, show that for both *Base-wo-Compressor* and *Base-full* model, adding Valpal database knowledge improved its performance on all three target languages. Furthermore, for all three target-languages, the improvement in performance of both *Base-wo-Compressor* and *Base-*

Model	it	de	zh	avg
Bootstrap	51.7	55.2	58.4	55.1
CModel	55.5	57.0	61.1	57.9
MAN-MOE	57.1	64.0	64.7	61.9
MoE	56.7	63.2	65.2	61.7
PGN	57.9	65.3	65.9	63.0
Base-wo-Compressor	37.1	49.7	45.3	44.0
Base-wo-Compressor+ Valpal	37.8	54.2	49.9	47.3
Increase	0.7	4.5	4.6	3.3
Base-full	57.2	65.1	68.8	63.7
Base-full+ Valpal	57.9	69.5	73.4	66.9
Increase	0.7	4.4	4.6	3.2

Table 3: Results for Monoloingual settings (with extended vocab for de and zh)

full models due to Valpal knowledge injection are same i.e 0.7 for *it*, 4.5 for *de* and 4.6 for *zh* (average 3.3). This provides the evidence that the improvement is indeed due to the Valpal Knowledge injection.

Model	it	de	zh	en	avg
MAN-MOE	57.7	66.2	65.9	66.0	63.9
MoE	57.1	63.5	66.1	64.1	62.7
PGN	58.0	65.7	66.9	67.8	64.6
Base-wo-Compressor	37.6	50.2	48.9	49.9	46.6
Base-wo-Compressor + Valpal	38.5	54.7	53.6	54.8	50.4
Increase	0.9	4.5	4.7	4.9	3.8

Table 4: Results in Polygot settings

6.2 Polyglot training

Table 4, outlines the results obtained under the polyglot training settings. For each experiment within these settings, the models are trained on a joint polyglot corpus of the three out of four languages namely *en*, *it*, *de* and *zh*, excluding the target language for which the results are outlined. For each experiment within these settings, the training corpus size is always fixed to 600,000 tokens to ensure controlled experiment-settings. We created such polyglot corpus by randomly sampling sentences from UPB train-set for each of the three source-languages until the token-size becomes approximately equal to 100,000, concatenated all these sampled datasets and randomly shuffled the order. *Alignment-projection* based approaches and the *Base-full* are not evaluated in the polyglot settings as these approaches require parallel-aligned source and target language sentence-pairs.

Results show that adding Valpal knowledge improves the performance of *Base-wo-Compressor* model, even within the polyglot settings, Furthermore, it is observed that although *Base-wo-Compressor* model performs better in polyglot training settings as compared to monolingual settings for most of the target languages, the improvement in performance of *Base-wo-Compressor* due to Valpal knowledge injection is same is both settings. This is because the fine-tuning of model with Valpal database knowledge is performed only with the unlabelled target-language corpus.

	it	de	zh
Vocab	125	128	122
Ext-vocab	–	975	415
Base-full	57.2	65.1	68.8
Base-full+ ValPal	57.9	65.9	68.7
Increase	0.7	0.8	0.9
Base-full+ ValPal-ext	–	69.5	73.4
Increase	0.7	4.4	4.6

Table 5: Results with and without ext-vocab

6.3 Performance with extended vocabularies

It can be observed in Tables 3 and 4 that the improvement on target-language is much lower than the improvements observed on zh, de and en. The reason being that we extended the Valpal vocabulary of en, zh and de using English Framenet (Barkley), Chinese Framenet (Yang et al., 2018) and German Framenet (of Texas) by the process described in section 2.4. However Italian Framenet is not publicly available.

We indeed performed experiments to analyze the impact of vocabulary extension on the performances. Table 5 outlines the results of these experiments. It can be observed in the table that extending the vocabulary of Valpal with the Framenet does lead to significant improvement in performance.

7 Conclusion

Valency Patterns Leipzig (ValPal) is a multilingual lexical database which provides the knowledge about the argument-patterns of various verb-forms in multiple languages including numerous low-resource languages. The database is originally created by the linguistic community to study the similarities and differences in the verb-patterns for various world's languages. In this work we utilised this database to improve the performance of the state-of-the-art cross-lingual model for SRL task.

We evaluated a framework to integrate the entire Valpal knowledge about any low-resource target-language into an LSTM based model. Our proposed framework only requires an unannotated target language corpus for the knowledge integration.

References

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating high quality proposition banks for multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2017. Transferring semantic roles using translation and syntactic information. *arXiv preprint arXiv:1710.01411*.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2019a. Cross-lingual transfer of semantic roles: From raw text to semantic roles. *arXiv preprint arXiv:1904.03256*.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2019b. Cross-lingual transfer of semantic roles: From raw text to semantic roles. *arXiv preprint arXiv:1904.03256*.

ICSI Barkley. English framenet.

Emanuele Bastianelli, Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2013. Textual inference and meaning representation in human robot interaction. In *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, pages 65–69.

Rui Cai and Mirella Lapata. 2020a. Alignment-free cross-lingual semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3883–3894.

Rui Cai and Mirella Lapata. 2020b. Alignment-free cross-lingual semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3883–3894.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2018. Multi-source cross-lingual model transfer: Learning what to share. *ACL*.

Janara Christensen, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pages 113–120.

Angel Daza and Anette Frank. 2019. Translate and label! an encoder-decoder approach for cross-lingual semantic role labeling. *arXiv preprint arXiv:1908.11326*.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Hao Fei, Meishan Zhang, and Donghong Ji. 2020. Cross-lingual semantic role labeling with high-quality translated training corpus. *arXiv preprint arXiv:2004.06295*.

Jiang Guo, Darsh J Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. *EMNLP*.

Jan Hajic, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, and Maria Antonia Martı. 2009. Lluıs marquez, adam meyers, joakim nivre, sebastian padó, jan štepánek, pavel stranák, mihai surdeanu, nianwen xue, and yi zhang. 2009. the conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18.

Iren Hartmann, Martin Haspelmath, and Bradley Taylor, editors. 2013. *The Valency Patterns Leipzig online database*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017a. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017b. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 473–483.

Atif Khan, Naomie Salim, and Yogan Jaya Kumar. 2015. A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing*, 30:737–747.

Philipp Koehn et al. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Mikhail Kozhevnikov and Ivan Titov. 2013. Cross-lingual transfer of semantic role labeling models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1200.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Ryan McDonald and Joakim Nivre. 2013. Yvonne irmbach-brundage, yoav goldberg, dipanjan das, kuzman ganchev, keith hall, slav petrov, hao zhang, oscar täckström, claudia bedini, núria bertomeu castelló, and jungmee lee. universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 92–97.

Todd K Moon. 1996. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60.

University of Texas. German framenet.

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Jacob Persson, Richard Johansson, and Pierre Nugues. 2009. Text categorization using predicate-argument structures. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 142–149.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 12–21.

Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Greedy, joint syntactic-semantic parsing with stack lstms. *arXiv preprint arXiv:1606.08954*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Hai Wang and Hoifung Poon. 2018. Deep probabilistic logic: A unifying framework for indirect supervision. *arXiv preprint arXiv:1808.08485*.

Nan Wang, Jiwei Li, Yuxian Meng, Xiaofei Sun, and Jun He. 2021. An mrc framework for semantic role labeling. *arXiv preprint arXiv:2109.06660*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Bright Xu. 2019. Nlp chinese corpus: Large scale chinese corpus for nlp.

Tsung-Han Yang, Hen-Hsen Huang, An-Zi Yen, and Hsin-Hsi Chen. 2018. Transfer of frames from english framenet to construct chinese framenet: a bilingual corpus-based approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137.