Transfer Learning for Low-Resource Part-of-Speech Tagging

Jeffrey Zhou  
Yale University

Neha Verma  
Yale University

Follow this and additional works at: https://elischolar.library.yale.edu/yurj

Part of the Computer Sciences Commons

Recommended Citation
Zhou, Jeffrey and Verma, Neha (2020) "Transfer Learning for Low-Resource Part-of-Speech Tagging," The Yale Undergraduate Research Journal: Vol. 1 : Iss. 1 , Article 19. Available at: https://elischolar.library.yale.edu/yurj/vol1/iss1/19

This Article is brought to you for free and open access by EliScholar – A Digital Platform for Scholarly Publishing at Yale. It has been accepted for inclusion in The Yale Undergraduate Research Journal by an authorized editor of EliScholar – A Digital Platform for Scholarly Publishing at Yale. For more information, please contact elischolar@yale.edu.
Transfer Learning for Low-Resource Part-of-Speech Tagging

Jeffrey Zhou†, Neha Verma†*
†Yale University
*These authors contributed equally to this work

Abstract

Neural network approaches to Part-of-Speech tagging, like other supervised neural network tasks, benefit from larger quantities of labeled data. However, in the case of low-resource languages, additional methods are necessary to improve the performances of POS taggers. In this paper, we explore transfer learning approaches to improve POS tagging in Afrikaans using a neural network. We investigate the effect of transferring network weights that were originally trained for POS tagging in Dutch. We also test the use of pretrained word embeddings in our POS tagger, both independently and in conjunction with the transferred weights from a Dutch POS tagger. We find a marginal increase in performance due to transfer learning with the Dutch POS tagger, and a significant increase due to the use of either unaligned or aligned pretrained embeddings. Notably, there is little difference in performance when using either unaligned or aligned embeddings, even when utilizing cross-lingual transfer learning.

1. INTRODUCTION

Part-of-Speech (POS) tagging is the task of tagging a word from text with its corresponding part-of-speech. The parts of speech are generally defined from a set of predefined tags. Current neural network approaches to this supervised learning task have been shown to perform at the same accuracy as human judgement in some cases, as in the case of the English Penn Treebank dataset (Bohnet et al., 2018; Akbik et al., 2018). However, the Penn Treebank is data-rich in terms of a treebank; neural networks are able to learn POS tagging with such high accuracy in part due to the sheer size of the corpus. The Penn Treebank contains about 40,000 tagged sentences (Marcus et al., 1993).

In comparison, the amount of data available in some other languages is far less. For example, in the Universal Treebank dataset, which is a Treebank available with data in 90 languages for dependency parsing and POS tagging, there are only 1,934 tagged sentences available in Afrikaans, a language spoken in South Africa (Augustinus et al., 2016). The low-resource nature of some languages makes neural network approaches to NLP tasks perform much worse than languages with more text resources, as the network is unable to learn these complex mappings with limited training resources.

Previous work on POS tagging for low-resource languages has exploited cross-lingual resources such as parallel data or bitext in order to transfer mappings from a data-rich language to low-resource language (Kim et al., 2015). Other approaches have used dictionaries to constrain the set of tags a word might appear as (Wisniewski et al., 2014). However, our work makes no use of parallel corpora or dictionaries. We simply transfer the knowledge from a POS tagger in a data-rich
language to a POS tagger for a related language with scarcer data. We use techniques from the field of transfer learning in order to complete this transfer of knowledge between languages.

In this paper, we create a POS tagging model for the low-resource language Afrikaans. We compare and combine cross-lingual and cross-domain transfer learning techniques, in the form of transferring model parameters and pretrained word embeddings, respectively. We use Dutch as a high-resource language that is closely related to Afrikaans for our cross-lingual transfer techniques.

We provide some background on common natural language processing (NLP) techniques used in this paper in section 2. Section 3 describes the model we created for our tagging task, including a description of the transfer process we performed to create our improved tagger. Section 4 describes the training process that we performed on our model, including the corpora used and parameters chosen. Section 5 includes the results of our model. Section 6 discusses the performance of our model and promising alternatives to our model, and section 7 summarizes our contribution.

2. RELATED WORK

2.1 Transfer Learning in NLP

Transfer Learning involves transferring knowledge learned from one task in order to improve performance on a related task (Torrey and Shavlik, 2010). Transfer learning is almost ubiquitous in the field of NLP at this point; almost all modern systems make use of pretrained embeddings for word representations. The use of embeddings is in fact a form of transfer learning; we leverage the results of language modelling on large datasets in order to improve the representations needed to perform tasks with smaller quantities of data. This form of transfer learning would be considered cross-domain as the tasks being completed differ, but the base language is the same (Ruder, 2019). Pretrained language modelling is a popular component in current models because language modelling does not require human annotation or labelled data of any sort, so the corpora upon which language models can be trained tend to be much larger (Ruder et al., 2019).

In the case of cross-lingual transfer learning, we leverage the relatedness of different languages to transfer task knowledge in one language to the same task in a different language. This kind of transfer learning has been shown to improve neural machine translation (Zoph et al., 2016), question answering (Lee and Lee, 2019), named entity recognition (Johnson et al., 2019), and other NLP tasks. This kind of transfer learning often consists of training a model for a specific task on a much larger dataset, and then fine tuning the model for the same task, but on a smaller dataset.

2.2 Pretrained Word Embedding

Pretrained embeddings are another form of transfer learning, in which word embeddings that have been learned for one task are then used for a similar task. The benefits of pretrained embeddings are twofold. Firstly, the use of pretrained embeddings has been shown to greatly improve the performance of neural network models for NLP tasks such as text classification (Ma and Hovy, 2016; Kim, 2014). This is likely because pretrained embeddings are trained on very large datasets, and thus contain better...
representations of rarely-occurring words when compared to training from scratch on sparser datasets.

The second benefit of using pretrained embeddings is that it is much more time-efficient than training from scratch. In order to obtain an accurate representation of the syntactic and semantic relationships among different words, a large number of parameters are necessary. Training these parameters from scratch is time consuming, especially when using large datasets necessary for the accuracy of the embeddings.

When working with multiple different languages, it is ideal to use aligned word embeddings (Joulin et al., 2018). This ensures that words from different languages can be compared in the same vector space. We anticipate that this will be particularly important in the context of a transfer learning neural network model; this will ensure that each of the embedding nodes of the two languages, as well as their associated downstream weights, are equivalent.

3. MODEL

3.1 Network

We choose a bidirectional LSTM (long short-term memory) network as our underlying model (Hochreiter and Schmidhuber, 1997). We choose this model in part because it is a strong choice for sequential tasks, like POS tagging, in which the input and output are of the same dimension. It is suitable for sequential tasks as it is a recurrent neural network; it incorporates information from previous inputs to the network in order to predict future states. We choose a bidirectional LSTM due to its ability to incorporate information both preceding and succeeding a state; this approach proves helpful in modelling natural language dependencies. Part-of-speech tagging requires the successful incorporation of a word's context in order to predict its part of speech. This is particularly important for words that have multiple meanings. Therefore, as the LSTM has strong capability to capture long-distance dependencies, we choose this as our baseline model for POS tagging. In fact, the baseline model underlying current state-of-the-art POS taggers is a bidirectional LSTM (Bohnet et al., 2018; Akbik et al., 2018).

The layers in our BiLSTM include an embedding layer, an LSTM layer, and a fully-connected linear layer. The embedding layer maps from the size of the language vocabulary to an embedding dimension. The LSTM, which consists of either 1 or 2 layers, as both models are tested in our experiments, maps from the embedding dimension to a hidden dimension (accounting for bidirectionality). Finally, our linear layer maps from the hidden dimension to the size of the tagset. For our baseline model, we allow PyTorch, a standard tool in deep learning, to randomly initialize our layers according to its presets for each of the layer types.

3.2 Transfer Learning Method

In incorporating transfer learning into our model for Afrikaans POS tagging, we adapt the method proposed in Zoph et al. (2016). The authors performed neural machine translation (NMT) between source languages Hausa, Turkish, Uzbek, and Urdu, and the target language English. They also trained an NMT model between French and English. With the greater availability of French-English bitext for an NMT system, the authors leveraged the performance of this system to improve the lower-resourced systems. To transfer the domain, NMT, knowledge from one system to another, the authors simply initialized the weights and biases in the low-resource model with the weights learned from the higher resource model. The idea of this transfer is that the eventual learned weights of the NMT system between the lower-resource language and English might resemble the weights of the French-English system more closely rather than some random initialization. A similar method is employed in Kocmi and Bojar (2018) where the authors train an NMT model between a high-resource language and target
language, and then continue training the model between a lower-resource language and the same target language, based on where the original training left off.

We employ this method by saving all weights and biases from the LSTM layer(s) and final linear layer of our high-resource POS tagger, and then instantiating our low-resource POS tagger with these weights and biases. We do not, however, transfer the weights learned from the embedding layer. The embedding layer maps from the size of the language's vocabulary to the given input size, and the size of the two language's vocabulary may differ. Therefore, we initialize the weight matrix in the embedding layer randomly in our transfer process.

### 3.3 Pretrained Word Embedding

We incorporated both unaligned and aligned pretrained embeddings from fastText into our model as well. These vectors were trained on Common Crawl and Wikipedia datasets. For Afrikaans, this consisted of 160 MB and 103 MB of data, respectively (Wenzek et al., 2019). Dutch has substantially more CommonCrawl data - it is about 200 times the size of Afrikaans. Dutch has about 22 times the number of Wikipedia articles as Afrikaans as well. The unaligned embeddings were obtained using a modified version of a skip-gram model, which generates word embeddings to predict context words given a central focal word (Bojanowski et al., 2016). Formally, the model aims to maximize the following log-likelihood:

$$\sum_{t=1}^{T} \sum_{c \in C_t} \log p(w_c | w_t)$$

where $C_t$ is a context window of fixed size around $w_t$, for a sequence of $T$ words $w_1, ..., w_T$. The modification introduces a subword model, where each word is represented as a bag of character SnS-grams, padded with boundary symbols < and > at the beginning and end of the word. For example, using the word chair and $n = 3$, we would get the following $n$-grams:

<ch, cha, hai, air, ir>

Note that with the boundary symbols, the n-gram 'air' is distinct from the word <air>. Each SnS-gram is then given its own vector embedding. A word is thus represented as the sum of the vector representations of its n-grams. This allows our model to learn patterns that appear across similar words. Furthermore, it allows our model to create embeddings for rare or previously unseen words, based on the n-grams that it has already seen.

The aligned versions of these vectors were generated using relaxed cross-domain similarity local scaling (RCSLS) (Joulin et al., 2018). Typically, alignment proceeds by learning an orthogonal linear mapping $W$ between the d-dimensional word embeddings of two languages, based on $n$ pairs of training words. The mapping minimizes a discrepancy measure between the two vectors, namely:

$$\min_{W \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i=1}^{n} \ell(Wx_i, y_i)$$

where $\ell$ is a loss function and $x$ and $y$ are the embeddings of the training words of each language. The ultimate goal is to create a linear mapping that extends beyond the SnS pairs of training words to all SnS source words. The RCSLS method follows the CSLS criterion as a loss function, which gives rise to the following optimization problem:

$$\min_{W \in \mathcal{O}_d} \left( \frac{1}{n} \sum_{i=1}^{n} -2x_i^T W^T y_i + \frac{1}{k} \sum_{y_i \in \mathcal{N}_k(x)} x_i^T W^T y_i \right)$$

where $\mathcal{O}_d$ is the set of orthogonal $d \times d$ matrices, and $\mathcal{N}_k(x)$ is the set of $k$ nearest neighbours in the set of target word vectors $Y = \{y_1, ..., y_k\}$. Typically, the use of an orthogonal weight matrix preserves the distances between word vectors, and thus their similarities. However, RCSLS does not strictly enforce
orthogonality; instead, it further introduces a formulation for relaxing this constraint. Rather than the set $O_d$, RCSLS seeks to minimize the above function over the convex hull of $O_d$.

Despite the non-orthogonal mapping, RCSLS has been reported to perform better than other methods of alignment (Joulin et al., 2018).

In order to incorporate these embeddings, we ensured that the embedding size of our model was the same as that of the pretrained word embeddings. For each word in our vocabulary (in any of the data splits), we first checked to see if an embedding for that word was already present. If so, we initialized our model with those weights for the embedding layer. Otherwise, we randomly initialized the weights for that word as described in section 3.2. Due to the extensive training corpora used for the fastText embeddings, all of the words in our vocabulary were already present in the pretrained embedding data.

4. EXPERIMENT

4.1 Corpora

We performed our experiments using the Dutch Alpino Treebank (Noord, 2002), and the Afrikaans AfriBooms Treebank (Augustinus et al., 2016), both of which were found within the Universal Dependencies dataset. The universal dependencies tagset includes 17 POS, shared across all languages. We use the given data splits within each treebank. We summarize the sizes of these datasets in Table 1.

<table>
<thead>
<tr>
<th>Treebank</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AfriBooms</td>
<td>1315</td>
<td>194</td>
<td>425</td>
<td>1934</td>
</tr>
<tr>
<td>Dutch</td>
<td>12264</td>
<td>718</td>
<td>596</td>
<td>13578</td>
</tr>
</tbody>
</table>

Table 1: Number of sentences in each split of datasets used

We use the lowercase forms of words for our experiments. Additionally, we prepend a beginning-of-sequence token to each sentence, and append an end-of-sentence token.

4.2 Evaluations

We To evaluate our model, we simply compute the accuracy per token in our test set. We take the index of the maximum value in the output vector as the predicted tag. We exclude padding tokens in our accuracy. We run our models with various modifications 5 times each.

4.3 Tested Modifications

We assess the accuracy of our model both with or without transferred weights from a Dutch POS tagger. In either of these two cases, we also tested the effect with unaligned pretrained embeddings, aligned pretrained embeddings, or with no pretrained embeddings at all, for a total of six experimental conditions.

5. RESULTS

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy, 1 Layer</th>
<th>Accuracy, 2 Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Afrikaans</td>
<td>89.67 (SD = 0.75)</td>
<td>89.93 (SD = 0.71)</td>
</tr>
<tr>
<td>Baseline + Unaligned Embeddings</td>
<td>92.35 (SD = 0.11)</td>
<td>92.49 (SD = 0.09)</td>
</tr>
<tr>
<td>Baseline + Aligned Embeddings</td>
<td>93.53 (SD = 0.13)</td>
<td>93.13 (SD = 0.36)</td>
</tr>
<tr>
<td>Dutch Transfer</td>
<td>90.32 (SD = 0.20)</td>
<td>90.50 (SD = 0.60)</td>
</tr>
<tr>
<td>Dutch Transfer + Unaligned Embeddings</td>
<td>93.05 (SD = 0.11)</td>
<td>93.21 (SD = 0.34)</td>
</tr>
<tr>
<td>Dutch Transfer + Aligned Embeddings</td>
<td>93.13 (SD = 0.37)</td>
<td>92.79 (SD = 0.17)</td>
</tr>
</tbody>
</table>

Table 2: Performances of various modifications to the Baseline Afrikaans POS Tagger. Means and standard deviations are reported for 5 runs.

We summarize the results of our model and its various modifications in Table 2. Accuracies and standard deviations
6. DISCUSSION

6.1 Experimental Results

Using transfer learning from Dutch appears to have marginally increased the accuracy of our Afrikaans model in most cases. In our single-layer and double-layer models using aligned embeddings, however, transferring knowledge from the Dutch POS tagger seemed to not help the accuracy of the Afrikaans tagger. This could be due in part to the alignment process used to transform the embeddings. Perhaps the alignment process allowed the Afrikaans embeddings to better suit the task of POS tagging, and/or lowered the ability of the Dutch embeddings to improve a Dutch POS tagger substantially, or at least enough to see an improvement in the Afrikaans tagger with transferred Dutch knowledge. In the case without any embeddings, there is a reliable increase in accuracy after using transferred Dutch knowledge, suggesting that the use of cross-language transfer learning in this domain is viable.

The use of both unaligned and aligned pretrained embeddings, however, seems to have greatly improved the performance of our model. Interestingly, the aligned embeddings appear to be more effective than the unaligned embeddings without transfer learning. This is surprising because unaligned embeddings are trained exclusively on the source language, whereas aligned embeddings are standardized across languages. This suggests that the RCSLS may not affect the quality of the word embeddings for monolingual tasks.

On the other hand, when utilizing a transfer learning approach, the success of the aligned embeddings was comparable to that of the unaligned embeddings. This was again surprising, because we anticipated that the aligned embeddings would benefit a cross-lingual approach such as transfer learning. This lack of difference between embedding alignments might be an artifact of the strong similarity between Dutch and Afrikaans. The embedding spaces of Dutch and Afrikaans may already be closely aligned, and performing the RCSLS algorithm may sacrifice some of the representative quality of the vectors in multilingual space in order to fit the alignment constraints.

6.2 Considerations

When using pretrained embeddings, there was the option to allow our model to train the weights of the embedding layer (as opposed to leaving them as fixed weights based on the pretrained embeddings). Preliminary testing gave significantly worse results when leaving the weights fixed. This is likely because the pretrained embeddings were trained for a different task, and thus cannot be expected to perform well when directly adapted for POS tagging. As such, for all subsequent experiments, we allowed our model to continuously train the weights of the embedding layer.

Another option for aligned pretrained word embeddings would be to use those aligned by the multilingual unsupervised and supervised embeddings (MUSE) package. The Facebook research team provides another source of aligned embeddings for multilingual use. Like the embeddings we used, the unaligned versions of these embeddings were originally trained using fastText. However, these embeddings appear to be a lot sparser than the ones we ultimately used; many words in our vocabulary were not present in the MUSE embeddings, leading to much worse performance in preliminary tests. Furthermore, embeddings aligned with RCSLS has been reported to perform better than those aligned with MUSE in NLP tasks such as machine translation (Joulin et al., 2018). This further supports our decision to use pretrained embeddings aligned by RCSLS.
6.3 Future Directions: Embeddings

We one possibility in the future is to explore pretrained word embeddings obtained from different sources. Unlike the prediction-base embeddings of fastText, the global vector model (GloVe) learns word embeddings from a $V \times V$ co-occurrence matrix (Pennington et al., 2014). Each cell of the co-occurrence matrix contains the number of times the two words occur together within a context window of a fixed size. GloVe typically utilizes context windows larger in size than prediction-based models like fastText, and is thus better suited to capturing longer-range dependencies. However, GloVe does not take into account the order of these dependencies. It would be interesting to explore whether GloVe pretrained embeddings are more effective than those trained by fastText.

Bidirectional encoder representations from transformers (BERT) are another method for generating word embeddings (Devlin et al., 2018). BERT utilizes a multilayer bidirectional transformer-encoder, and is trained on two unsupervised tasks. The first is a masked language model, where a percentage of words are replaced with a [MASK] token. The second is a “next sentence prediction” task, where the network has to classify if a sentence follows a given sentence or not. Many current state-of-the-art NLP models utilize word embeddings learned in the BERT model. Multilingual BERT supports 100 languages, and partially aligned word embeddings can be extracted for each of these languages (Cao et al., 2020).

6.4 Future Directions: Language Pairs

Besides embeddings, another direction we could have explored is the set of languages we tested upon. We chose Dutch and Afrikaans as they were a strong exemplar of related languages in which one of the languages was resource scarce. However, Dutch and Afrikaans are quite strongly related, as Afrikaans is a direct daughter language of Dutch, and many of the words in the vocabulary are shared with Dutch. However, their morphology and grammar do differ. We noted that in testing the use of aligned multilingual embeddings for cross-lingual transfer learning, we did not see an improvement in performance, in comparison with the use of unaligned embeddings. We hypothesized that this was an artifact of the strong lexical overlap between Dutch and Afrikaans. Therefore, in order to more fully test the use of aligned embeddings in this application of cross-lingual transfer learning, we could try this same approach on a set of languages that are more distantly related. In such cases, there may be a greater need for the use of aligned embeddings as the vector representations of more distantly related languages may be more distant. We predict that aligned embeddings may make the parameters more interpretable between languages as the low-resource language adopts the higher-resource language parameters as its initialization.

We could have also used a variety of high-resource languages to provide the network initialization. In testing a variety of language pairs, we could have determined which languages are most suited to for cross-lingual transfer learning for the target low-resource language. We predict that the degree of language relatedness plays a large role in the effectiveness of cross-lingual transfer. In such an investigation, we could have also computed confusion matrices before and after transfer learning occurs in order to determine if different high-resource languages transfer knowledge of parts of speech in different ways. We note languages within the same language family, according to the Universal Dependencies treebank, as a good example of testing both more distant languages, as well as different high-resource languages. For example, we could imagine creating a POS tagger for Belarusian, a low-resource language, and testing different Slavic languages, like Russian, Polish, and Ukrainian, as high-resource languages for transfer.
7. CONCLUSION

We introduced improvements to a baseline BiLSTM POS tagger for a low-resource language, Afrikaans. We introduced cross-lingual transfer learning to improve this model by training a Dutch POS tagger, and instantiating the parameters of the Afrikaans POS tagger with the parameters learned from the Dutch model. We also investigated the use of multilingual embeddings, both aligned and unaligned, and their ability to improve our transfer model. Our model showed variable performance with the introduction of transfer learning via specific weight initialization, but consistent improvement with the introduction of multilingual word embeddings. We note small difference between the use of aligned and unaligned embeddings. We observe the best performance of our model with the use of unaligned embeddings and specific weight initialization.

Despite to the marginal increases in accuracy observed in our experiments thus far, these improvements nonetheless point us towards concrete new directions to explore. Given the similarities between aligned and unaligned embeddings, we hope to investigate the effect of embeddings trained from other models such as GloVe or BERT. Furthermore, the effectiveness of our strategies in improving Afrikaans POS tagging raises the question of whether similar improvements would also be seen in other language pairs. Though our approach is still rudimentary, it demonstrates the feasibility of utilizing transfer learning and pretrained word embeddings for improving linguistic tasks in low-resource languages.

8. ENDNOTES


Bernd Bohnet, Ryan McDonald, Goncalo Simoes, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings.


Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations.


Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In EMNLP.


