Reducing Confusion in Active Learning for Part-Of-Speech Tagging

Aditi Chaudhary[1], **Antonios Anastasopoulos**[2,*], **Zaid Sheikh**[1], **Graham Neubig**[1]
[1]Language Technologies Institute, Carnegie Mellon University
[2]Department of Computer Science, George Mason University
{aschaudh,zsheikh,gneubig}@cs.cmu.edu antonis@gmu.edu

Abstract

Active learning (AL) uses a data selection algorithm to select useful training samples to minimize annotation cost. This is now an essential tool for building low-resource syntactic analyzers such as part-of-speech (POS) taggers. Existing AL heuristics are generally designed on the principle of selecting *uncertain* yet *representative* training instances, where annotating these instances may reduce a large number of errors. However, in an empirical study across six typologically diverse languages (German, Swedish, Galician, North Sami, Persian, and Ukrainian), we found the surprising result that even in an *oracle* scenario where we know the true uncertainty of predictions, these current heuristics are far from optimal. Based on this analysis, we pose the problem of AL as selecting instances that *maximally reduce the confusion between particular pairs of output tags*. Extensive experimentation on the aforementioned languages shows that our proposed AL strategy outperforms other AL strategies by a significant margin. We also present auxiliary results demonstrating the importance of proper calibration of models, which we ensure through cross-view training, and analysis demonstrating how our proposed strategy selects examples that more closely follow the oracle data distribution. The code is publicly released here.[1]

1 Introduction

Part-of-speech (POS) tagging is a crucial step for language understanding, both being used in automatic language understanding applications such as named entity recognition (NER; Ankita and Nazeer, 2018) and question answering (QA; Wang et al., 2018), but also being used in *manual* language understanding by linguists who are attempting to answer linguistic questions or document less-resourced languages (Anastasopoulos et al., 2018). Much prior work (Huang et al., 2015; Bohnet et al., 2018) on developing high-quality POS taggers uses neural network methods, which rely on the availability of large amounts of labelled data. However, such resources are not readily available for the majority of the world's 7000 languages (Hammarström et al., 2018). Furthermore, manually annotating large amounts of text with trained experts is an expensive and time-consuming task, even more so when linguists/annotators might not be native speakers of the language.

Active Learning (Lewis, 1995; Settles, 2009, AL) is a family of methods that aim to train effective models with less human effort and cost by selecting such a subset of data that maximizes the end model performance. Although many methods have been proposed for AL in sequence labeling (Settles and Craven, 2008; Marcheggiani and Artières, 2014; Fang and Cohn, 2017), through an empirical study across six typologically diverse languages we show that within the same task setup these methods perform inconsistently. Furthermore, even in an *oracle* scenario, where we have access to the true labels during data selection, existing methods are far from optimal.

We posit that the primary reason for this inconsistent performance is that while existing methods consider uncertainty in predictions, they do not consider the *direction* of the uncertainty with respect to the output labels. For instance, in Figure 1 we consider the German token ''die,'' which may be either a pronoun (PRO) or determiner (DET). According to the initial model (iteration 0), ''die'' was labeled as PRO majority of the time, but a significant amount of probability mass was also assigned to other output tags (OTHER) for many examples. Based on this, existing AL algorithms that select uncertain tokens will likely select ''die'' because it is frequent and

*Work done at Carnegie Mellon University.
[1]https://github.com/Aditi138/CRAL.

Transactions of the Association for Computational Linguistics, vol. 9, pp. 1–16, 2021. https://doi.org/10.1162/tacl_a_00350
Action Editor: Yuji Matsumoto. Submission batch: 3/2020; Revision batch: 6/2020; Published 02/2021.

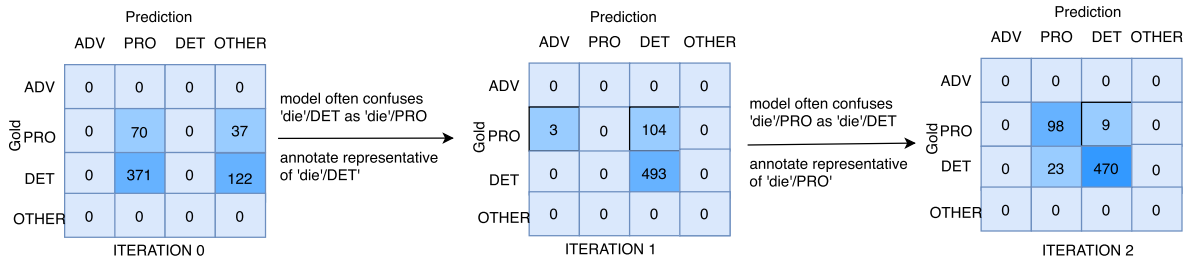

Figure 1: Illustration of selecting representative token-tag combinations to reduce confusion between the output tags on the German token ‘‘die’’ in an idealized scenario where we know true model confusion.

its predictions are not certain, but they may select an instance of ‘‘die’’ with *either* a gold label of PRO or DET. Intuitively, because we would like to correct errors where tokens with true labels of DET are mislabeled by the model as PRO, asking the human annotator to tag an instance with a true label of PRO, even if it is uncertain, is not likely to be of much benefit.

Inspired by this observation, we pose the problem of AL for POS tagging as selecting tokens that maximally *reduce the confusion* between the output tags. For instance, in the example, we would attempt to pick a token-tag pair ‘‘die/DET’’ to reduce potential errors of the model over-predicting PRO despite its belief that DET is also a plausible option. We demonstrate the features of this model in an oracle setting where we know true model confusions (as in Figure 1), and also describe how we can approximate this strategy when we do not know the true confusions.

We evaluate our proposed AL method by running simulation experiments on six typologically diverse languages, namely, German, Swedish, Galician, North Sami, Persian, and Ukrainian, improving upon models seeded with crosslingual transfer from related languages (Cotterell and Heigold, 2017). In addition, we conduct human annotation experiments on Griko, an endangered language that truly lacks significant resources. Our contributions are as follows:

1. We empirically demonstrate the shortcomings of existing AL methods under both conventional and ‘‘oracle’’ settings. Based on the subsequent analysis, we propose a new AL method that achieves +2.92 average per-token accuracy improvement over existing methods under conventional settings, and a +2.08 average per-token accuracy improvement under the *oracle* setting.

2. We conduct extensive analysis measuring how the selected data using our proposed AL method closely matches the oracle data distribution.

3. We further demonstrate the importance of model calibration, the accuracy of the model’s probability estimates themselves, and demonstrate that cross-view training (Clark et al., 2018) is an effective way to improve calibration.

4. We perform human annotation using the proposed method on an endangered language, Griko, and find our proposed method to perform better than the existing methods. In this process, we collect 300 new token-level annotations which will help further Griko NLP.

2 Background: Active Learning

Generally, AL methods are designed to select data based on two criteria: ‘‘informativeness’’ and ‘‘representativeness’’ (Huang et al., 2010). Informativeness represents the ability of the selected data to reduce the model uncertainty on its predictions, and representativeness measures how well the selected data represent the entire unlabeled data. AL is an iterative process and is typically implemented in a batched fashion for neural models (Sener and Savarese, 2018). In a given iteration, a batch of data is selected using some heuristic on which the end model is trained until convergence. This trained model is then used to select the next batch for annotation, and so forth.

In this work we focus on *token-level* AL methods, which require annotation of individual tokens in context, rather than full sequence annotation, which is more time consuming. Given an unlabeled pool of sequences $D = \{\mathbf{x_1}, \mathbf{x_2},$

$\cdots, \mathbf{x_n}\}$ and a model θ, $P_\theta(y_{i,t} = j \mid \mathbf{x_i})$ denotes the output probability of the output tag $j \in \mathcal{J}$ produced by the model θ for the token $x_{i,t}$ in the input sequence $\mathbf{x_i}$. $\mathcal{J}$ denotes the set of POS tags. Most popular methods (Settles, 2009; Fang and Cohn, 2017) define the ''informativeness'' using either *uncertainty sampling* or *query-by-committee*. We provide a brief review of these existing methods.

- **Uncertainty Sampling** (UNS; Fang and Cohn, 2017) selects the most uncertain word types in the unlabeled corpus D for annotation. First, it calculates the token entropy $H(x_{i,t}; \theta)$ for each unlabeled sequence $\mathbf{x_i} \in D$ under model θ, defined as

$$p_{i,t,j} := P_\theta(y_{i,t} = j \mid \mathbf{x_i})$$
$$H(x_{i,t}; \theta) = -\sum_{j \in \mathcal{J}} p_{i,t,j} \log p_{i,t,j}$$

  Next, this entropy is aggregated over all token occurrences across D to get an uncertainty score $S_{\text{UNS}}(z)$ for each word type z:

$$S_{\text{UNS}}(z) = \sum_{\mathbf{x_i} \in D} \sum_{x_{i,t} = z} H(x_{i,t}; \theta)$$

- **Query-by-Commitee** (QBC; Settles and Craven, 2008) selects the tokens having the highest disagreement between a committee of models $C = \{\theta_1, \theta_2, \theta_3, \cdots\}$, which is aggregated over all token occurrences. The token level disagreement scores are defined as

$$S_{\text{DIS}}(x_{i,t}) = |C| - \max \sum_{y \in [\hat{y}^{\theta_1}_{i,t}, \hat{y}^{\theta_2}_{i,t}, \cdots, \hat{y}^{\theta_c}_{i,t}]} V(y),$$

  where $V(y)$ is number of ''votes'' received for the token label y. $\hat{y}^{\theta_c}_{i,t}$ is the prediction with the highest score according to model θ_c for the token $x_{i,t}$. These disagreement scores are then aggregated over word types:

$$S_{\text{QBC}}(z) = \sum_{\mathbf{x_i} \in D} \sum_{x_{i,t} = z} S_{\text{DIS}}(x_{i,t})$$

Finally, regardless of whether we use an UNS-based or QBC-based score, the top b word types with the highest aggregated score are then selected as the to-label set

$$X_{\text{LABEL}} = \text{b-}\arg\max_z S(z),$$

where b-$\arg\max$ selects top b word types having the highest $S(z)$. Fang and Cohn (2017) and Chaudhary et al. (2019) further attempt to include the ''representativeness'' criterion by combining uncertainty sampling with a bias towards high-frequency tokens/spans.

Failings of Current AL Methods Although these methods are widely used, in a preliminary empirical study we found that these existing methods are less than optimal, and fail to bring consistent gains across multiple settings. Ideally, having a single strategy that performs the best across a diverse language set is useful for other researchers who plan to use AL for new languages. Instead of researchers experimenting with different strategies with human annotation, which is costly, having a single strategy known a priori will reduce both time and human annotation effort. Specifically, we demonstrate this problem of inconsistency through a set of *oracle* experiments, where the data selection algorithm has access to the true labels. We hope that these experiments serve as an upper-bound for their non-oracle counterparts, so if existing methods do not achieve gains even in this case, they will certainly be even less promising when true labels are not available at data selection time, as is the case in standard AL.

Concretely, as an oracle *uncertainty sampling* method UNS-ORACLE, we select types with the highest negative log likelihood of their true label. As an ''oracle'' QBC method QBC-ORACLE, we select types having the largest number of incorrect predictions. We conduct 20 AL iterations for each of these methods across six typologically diverse languages.[2]

First, we observe that between the oracle methods (Figure 2) no method consistently performs the best across all six languages. Second, we find that just considering uncertainty leads to unbalanced selection of the resulting tags. To drive this point across, Table 1 shows the output tags selected for the German token ''zu'' across multiple iterations. UNS-ORACLE selects the most frequent output tag, failing to select tokens from other output tags. Whereas QBC-ORACLE selects tokens having multiple tags, the distribution is not in proportion with the true tag distribution. Our hypothesis is that this inconsistent performance occurs because none of the methods consider the

[2]More details on the experimental setup in Section §5.

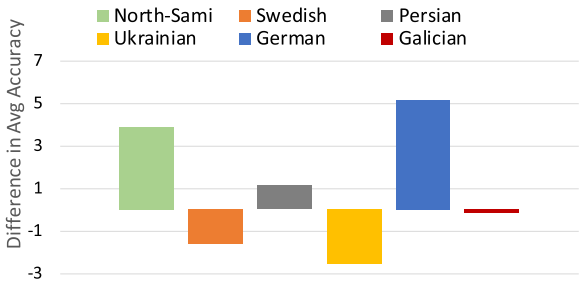

Figure 2: Illustrating the inconsistent performance of UNS-ORACLE and QBC-ORACLE methods. The y-axis is difference in the POS accuracy for these two methods, averaged across 20 iterations having a batch size 50.

	QBC-ORACLE	UNS-ORACLE
ITERATION-1	PART=1	ADP=1
ITERATION-2	PART=1,ADP=1	ADP=2
ITERATION-3	ADV=1,PART=1,ADP=1	ADP=2
ITERATION-4	ADV=1,PART=1,ADP=2	ADP=3

Table 1: Each cell is the tag selected for German token ‘‘zu’’ at each iteration. Gold output tag distribution for ‘‘zu’’ is ADP=194, PART=103, ADV=5, PROPN=5, ADJ=1.

confusion between output tags while selecting data. This is especially important for POS tagging because we find that the existing methods tend to select highly syncretic word types. *Syncretism* is a linguistic phenomenon where distinctions required by syntax are not realized by morphology, meaning a word type can have multiple POS tags based on context.[3] This is expected because syncretic word types, owing to their inherent ambiguity, cause high uncertainty, which is the underlying criterion for most AL methods.

3 Confusion-Reducing Active Learning

To address the limitations of the existing methods, we propose a *confusion-reducing active learning* (CRAL) strategy, which aims at reducing the confusion between the output tags. In order to combine both ‘‘informativeness’’ and ‘‘representativeness’’, we follow a two-step algorithm:

1. **Find the most confusing word types.** The goal of this step is to find b word types that would *maximally reduce* the model confusion within the output tags. For each token $x_{i,t}$ in the unlabeled sequence $\mathbf{x_i} \in D$, we first define the confusion as the sum of probability $P_\theta(y_{i,t} = j \mid \mathbf{x_i})$ of all output tags $\mathcal{J}$ other than the highest probability output tag $\hat{y}_{i,t}$:

$$S_{\text{CONF}}(x_{i,t}) = 1 - P_\theta(y_{i,t} = \hat{y}_{i,t} \mid \mathbf{x_i}),$$

then sum this over all instances of type z:

$$S_{\text{CRAL}}(z) = \sum_{\mathbf{x_i} \in D} \sum_{x_{i,t}=z} S_{\text{CONF}}(x_{i,t}).$$

Again selecting the top b types having the highest score (given by b-$\arg\max$) gives us the most confusing word types (X_{INIT}). For each token, we also store the output tag that had the second highest probability which we refer to as the ‘‘most confusing output tag’’ for a particular $x_{i,t}$:

$$O(x_{i,t}, j) = \begin{cases} 1 & \text{if } j = \arg\max_{j \in \mathcal{J} \setminus \{\hat{y}_{i,t}\}} p_{i,t,j} \\ 0 & \text{otherwise.} \end{cases}$$

For each word type z, we aggregate the frequency of the most confusing output tag across all token occurrences:

$$O_{\text{CRAL}}(z, j) = \sum_{\mathbf{x_i} \in D} \sum_{x_{i,t}=z} O(x_{i,t}, j),$$

and compute the output tag with the highest frequency as the most confusing output tag for type z:

$$L(z) = \arg\max_{j \in \mathcal{J}} O_{\text{CRAL}}(z, j).$$

For each of the top b most confusing word types, we retrieve its most confusing output tag, resulting in type-tag pairs given by $L_{\text{INIT}} = \{\langle z_1, j_1\rangle, \cdots \langle z_b, j_b\rangle\}$. This process is illustrated in steps 7–14 in Algorithm 1.

2. **Find the most representative token instances**. Now that we have the most confusing type-tag pairs L_{INIT}, our final step is selecting the most representative token instances for annotation. For each type-tag tuple $\langle z_k, j_k\rangle \in L_{\text{INIT}}$, we first retrieve contextualized representations for all token occurrences ($x_{i,t} = z_k$) of the word-type z_k from the encoder of the POS model. We express this in shorthand as $\mathbf{c}_{i,t} := \text{enc}(x_{i,t})$. Because the true labels are unknown, there is no certain way of knowing which tokens have the ‘‘most confusing output tag’’ as the true label. Therefore, each token representation

[3]Details can be found in Section §5.2, Table 3.

Algorithm 1 CONFUSION-REDUCING AL

1: $D \leftarrow$ *unlabeled set of sequences*
2: $Z \leftarrow$ *vocabulary*
3: $\mathcal{J} \leftarrow$ *output tag-set*
4: $b \leftarrow$ *active learning batch size*
5: $P_\theta(y_{i,t} = j \mid \mathbf{x_i}) \leftarrow$ *marginal probability*
6: $p_{i,t,j} := P_\theta(y_{i,t} = j \mid \mathbf{x_i})$
7: **for** $\mathbf{x_i} \in D$ **do**
8: **for** $(x_{i,t}) \in \mathbf{x_i}$ **do**
9: $z \leftarrow x_{i,t}$
10: $S_{\text{CRAL}}(z) \leftarrow S_{\text{CRAL}}(z) + (1 - p_{i,t,\hat{y}_{i,t}})$
11: $\hat{j} \leftarrow \arg\max_{j \in \mathcal{J} \setminus \{\hat{y}_{i,t}\}} p_{i,t,j}$
12: $O_{\text{CRAL}}(z, \hat{j}) \leftarrow O_{\text{CRAL}}(z, \hat{j}) + 1$
13: $X_{\text{INIT}} \leftarrow$ b-$\arg\max_{z \in Z} S_{\text{CRAL}}(z)$
14: **for** $z_k \in X_{\text{INIT}}$ **do**
15: $j_k \leftarrow \arg\max_{j \in \mathcal{J}} O_{\text{CRAL}}(z_k, j)$
16: **for** $x_{i,t} \in D$ s.t. $x_{i,t} = z_k$ **do**
17: $\mathbf{c}_{x_{i,t}} \leftarrow \text{enc}(x_{i,t})$
18: $W_{x_{i,t}} = p_{i,t,j_k} * \mathbf{c}_{x_{i,t}}$
19: $X_{\text{LABEL}}(z_k) = \text{CENTROID}\{W_{x_{i,t}=z_k}\}$

$\mathbf{c}_{i,t}$ is weighted with the model confidence of the most confusing tag j_k given by

$$W_{x_{i,t}} = P_\theta(y_{i,t} = j_k \mid \mathbf{x_i}) * \mathbf{c}_{i,t}$$

Finally, the token instance that is closest to the centroid of this weighted token set becomes the most representative instance for annotation. Going forward, we also refer to the most representative token instance as the centroid for simplicity.[4] This process is repeated for each of the word-types z_k, resulting in the to-label set X_{LABEL}. This is illustrated in steps 14–19 in Algorithm 1.

During the annotation process, the selected representative tokens of each selected confusing word type are presented in context similar to Fang and Cohn (2017) and Chaudhary et al. (2019).

[4]Sener and Savarese (2018) describe why choosing the centroid is a good approximation of representativeness. They pose AL as a core-set selection problem where a core set is the subset of data on which the model if trained closely matches the performance of the model trained on the entire dataset. They show that finding the core set is equivalent to choosing b center points such that the largest distance between a data point and its nearest center is minimized. We take inspiration from this result in using the centroid to be the most representative instance.

4 Model and Training Regimen

Now that we have a method to select data for annotation, we present our POS tagger in Section §4.1, followed by the training algorithm in Section §4.2.

4.1 Model Architecture

Our POS tagging model is a hierarchical neural conditional random field (CRF) tagger (Ma and Hovy, 2016; Lample et al., 2016; Yang et al., 2017). Each token $(\mathbf{x}, t)$ from the input sequence $\mathbf{x}$ is first passed through a character-level Bi-LSTM, followed by a self-attention layer (Vaswani et al., 2017), followed by another Bi-LSTM to capture information about subword structure of the words. Finally, these character-level representations are fed into a token-level Bi-LSTM in order to create contextual representations $\mathbf{c_t} = \overrightarrow{\boldsymbol{h_t}} : \overleftarrow{\boldsymbol{h_t}}$, where $\overrightarrow{\boldsymbol{h_t}}$ and $\overleftarrow{\boldsymbol{h_t}}$ are the representations from the forward and backward LSTMs, and ‘‘:’’ denotes the concatenation operation. The encoded representations are then used by the CRF decoder to produce the output sequence.

Because we acquire token-level annotations, we cannot directly use the traditional CRF, which expects a fully labeled sequence. Instead, we use a constrained CRF (Bellare and McCallum, 2007), which computes the loss only for annotated tokens by marginalizing the un-annotated tokens, as has been used by prior token-level AL models (Fang and Cohn, 2017; Chaudhary et al., 2019) as well. Given an input sequence $\mathbf{x}$ and a label sequence $\mathbf{y}$, traditional CRF computes the likelihood as follows:

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{\prod_{t=1}^{N} \psi_t(y_{t-1}, y_t, \mathbf{x}, t)}{Z(\mathbf{x})},$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y} \in \mathbf{Y}(N)} \prod_{t=1}^{N} \psi_t(y_{t-1}, y_t, \mathbf{x}, t),$$

where N is the length of the sequence, $\mathbf{Y}(N)$ denotes the set of all possible label sequences with length N. $\psi_t(y_{t-1}, y_t, \mathbf{x}) = \exp(\mathbf{W}^T_{y_{t-1}, y_t} \mathbf{x}_t + \mathbf{b}_{y_{t-1}, y_t})$ is the energy function where $\mathbf{W}^T_{y_{t-1}, y_t}$ and $\mathbf{b}_{y_{t-1}, y_t}$ are the weight vector and bias corresponding to label pair (y_{t-1}, y_t) respectively. In constrained CRF training, $\mathbf{Y_L}$ denotes the set of all possible sequences that are congruent with the observed annotations, and the likelihood is computed as: $p_\theta(\mathbf{Y_L}|\mathbf{x}) = \sum_{\mathbf{y} \in \mathbf{Y_L}} p_\theta(\mathbf{y}|\mathbf{x})$.

4.2 Cross-view Training Regimen

In order to further improve this model, we apply cross-view training (CVT), a semi-supervised learning method (Clark et al., 2018). On unlabeled examples, CVT trains auxiliary prediction modules, which look at restricted ''views'' of an input sequence, to match the prediction from the full view. By forcing the auxiliary modules to match the full-view module, CVT improves the model's representation learning. Not only does it help in improving the downstream performance under low-resource conditions, but also improves the model calibration overall (§5.4). Having a well-calibrated model is quite useful for AL, as a well-calibrated model tends to assign lower probabilities to ''true'' incorrect predictions, which allows the AL measure to select these incorrect tokens for annotation.

CVT is composed of four auxiliary prediction modules, namely: the forward module θ_{fwd} that makes predictions without looking at the right of the current token, the backward module θ_{bwd} that makes predictions without looking at the left of the current token, the future module θ_{fut} that does not look at either the right context or the current token, and the past module θ_{pst} that does not look at either the left context or the current token. The token representations $\mathbf{c_t}$ for each module can be seen as follows:

$$\mathbf{c_t^{fwd}} = \overrightarrow{\boldsymbol{h_t}}, \quad \mathbf{c_t^{bwd}} = \overleftarrow{\boldsymbol{h_t}}, \quad \mathbf{c_t^{full}} = \overrightarrow{\boldsymbol{h_t}} : \overleftarrow{\boldsymbol{h_t}}.$$
$$\mathbf{c_t^{fut}} = \overrightarrow{\boldsymbol{h_{t-1}}}, \quad \mathbf{c_t^{pst}} = \overleftarrow{\boldsymbol{h_{t+1}}}.$$

For an unlabeled sequence $\mathbf{x}$, the full-view model θ_{full} first produces soft targets $p_\theta(\mathbf{y}|\mathbf{x})$ after inference. CVT matches the soft predictions from V auxiliary modules by minimizing their KL-divergence. Although CRF produces a probability distribution over all possible output sequences, for computational feasibility we compute the token-level KL-divergence using $p_\theta(y_t|\mathbf{x})$, which is the marginal probability distribution of token $(\mathbf{x}, t)$ over all output tags T. This is calculated easily from the forward-backward algorithm:

$$L_{\text{CVT}} = \frac{1}{|D|} \sum_{\mathbf{x_i} \in D} \sum_{x_{i,t} \in \mathbf{x_i}} \sum_{v=1}^{V} KL(p_\theta^{full} || p_\theta^v)$$

$$p_\theta^{full} := P_\theta^{full}(y_{i,t}{=}j \,|\, \mathbf{x_i}) \text{ and } p_\theta^v := P_\theta^v(y_{i,t}{=}j|\mathbf{x_i})$$

where $|D|$ is the total unlabeled examples in D.

4.3 Cross-Lingual Transfer Learning

Using the architecture described above, for any given target language we first train a POS model on a group of related high-resource languages. We then *fine-tune* this pre-trained model on the newly acquired annotations on the target language, as obtained from an AL method. The objective of cross-lingual transfer learning is to warm-start the POS model on the target language. Several methods have been proposed in the past including annotation projection (Zitouni and Florian, 2008), and model transfer using pre-trained models such as m-BERT (Devlin et al., 2019). In this work our primary focus is on designing an active learning method, so we simply pre-train a POS model on a group of related high-resource languages (Cotterell and Heigold, 2017), which is a computationally cheap solution, a crucial requirement for running multiple AL iterations. Furthermore, recent work (Siddhant et al., 2020) has shown the advantage of pre-training using a selected set of related languages over a model pre-trained over all available languages.

Following this, for a given target language we first select a set of typologically related languages. An initial set of transfer languages is obtained using the automated tool provided by Lin et al. (2019), which leverages features such as phylogenetic similarity, typology, lexical overlap, and size of available data, in order to predict a list of optimal transfer languages. This list can be then refined using the experimenter's intuition. Finally, a POS model is trained on the concatenated corpora of the related languages. Similar to Johnson et al. (2017), a language identification token is added at the beginning and end of each sequence.

5 Simulation Experiments

In this section, we describe the simulation experiments used for evaluating our method. Under this setting, we use the provided training data as our unlabeled pool and simulate annotations by using the gold labels for each AL method.

Datasets: For the simulation experiments, we test on six typologically diverse languages: German, Swedish, North Sami, Persian, Ukrainian, and Galician. We use data from the Universal Dependencies (UD) v2.3 (Nivre et al., 2016; Nivre et al., 2018; Kirov et al., 2018) treebanks with the

TARGET LANGUAGE	TRANSFER LANGUAGES (TREEBANK)
German (de-gsd)	English (en-ewt) + Dutch (nl-alpino)
Swedish (sv-lines)	Norwegian (no-nynorsk) + Danish (da-ddt)
North Sami (sme-giella)	Finnish (fi-ftb)
Persian (fa-seraji)	Urdu (ur-udtb) + Russian (ru-gsd)
Galician (gl-treegal)	Spanish (es-gsd) + Portuguese (pt-gsd)
Ukrainian (uk-iu)	Russian (ru-gsd)
Griko	Greek (el-gdt) + Italian (it-postwita)

Table 2: Dataset details describing the group of related languages over which the model was pre-trained for a given target language.

same train/dev/test split as proposed in McCarthy et al. (2018). For each target language, the set of related languages used for pre-training is listed in Table 2. Persian and Urdu datasets being in the Perso-Arabic script, there is no orthography overlap along the transfer and the target languages. Therefore, for Persian we use uroman,[5] a publicly available tool for romanization.

Baselines: As described in Section §2, we compare our proposed method (CRAL) with *Uncertainty Sampling* (UNS) and *Query-by-commitee* (QBC). We also compare with a random baseline (RAND) that selects tokens randomly from the unlabeled data D. For QBC, we use the following committee of models $C = \{\theta_{fwd}, \theta_{bwd}, \theta_{full}\}$, where θ_i are the CVT views (§4.2). We do not include the θ_{fut} and θ_{pst} as they are much weaker in comparison to the other views.[6] For CRAL, UNS, and RAND, we use the full model view.

Model Hyperparameters: We use a hidden size of 25 for the character Bi-LSTM, 100 for the modeling layer, and 200 for the token-level Bi-LSTM. Character embeddings are 30-dimensional and are randomly initialized. We apply a dropout of 0.3 to the character embeddings before inputting to the Bi-LSTM. A further 0.5 dropout is applied to the output vectors of all Bi-LSTMs. The model is trained using the SGD optimizer with learning rate of 0.015. The model is trained till convergence over a validation set.

Active Learning Parameters: For all AL methods, we acquire annotations in batches of 50 and run 20 simulation experiments resulting in a total of 1000 tokens annotated for each method. We pre-train the model using the above parameters and after acquiring annotations, we fine-tune it with a learning rate proportional to the number of sentences in the labeled data $lr = 2.5e^{-5}|X_{\text{LABEL}}|$.

[5] https://www.isi.edu/~ulf/uroman.html.

[6] We chose CVT views for QBC over the ensemble for computational reasons. Training three models independently would require three times the computation. Given that for each language we run 20 experiments, amounting to a total of 120 experiments, reducing the computational burden was preferred.

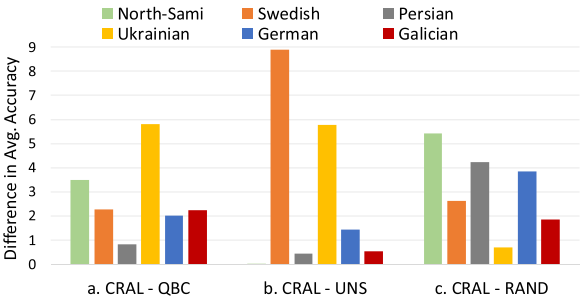

Figure 3: Our method (CRAL) outperforms existing AL methods for all six languages. y-axis is the difference in POS accuracy between CRAL and other AL methods, averaged across 20 iterations with batch size 50.

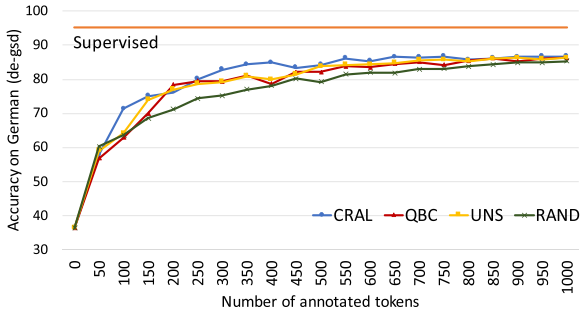

Figure 4: Comparison of the POS performance across the different methods for 20 AL iterations for German.

5.1 Results

Figure 3 compares our proposed CRAL strategy with the existing baselines. The y-axis represents the difference in POS tagging performance between two AL methods and is measured by accuracy. The accuracy is averaged across 20 iterations. Across all six languages, our proposed method CRAL shows significant performance gains over the other methods. In Figure 4 we plot the individual accuracy values across the 20 iterations for German and we see that our proposed method CRAL performs consistently better across multiple iterations. We also see that the zero-shot model on German (iteration-0) gets a decent warm start because of cross-lingual transfer from Dutch and English.

Furthermore, to check how the performance of the AL methods is affected by the underlying POS tagger architecture, we conduct additional

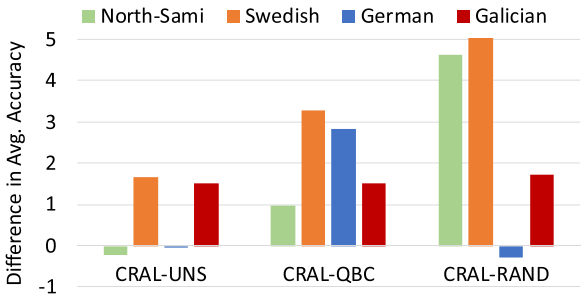

Figure 5: Comparing the difference in POS performance across the AL methods with BRNN/MLP architecture, averaged across 20 iterations.

TARGET LANGUAGE	UNS	QBC	CRAL
German	74 %	76 %	82 %
Swedish	56 %	54 %	62 %
North-Sami	10 %	12 %	14 %
Persian	50 %	46 %	46 %
Galician	40 %	42 %	44 %
Ukrainian	38 %	48 %	48 %

Table 3: Percentage of syncretic word types in the first iteration of active learning (consisting of 50 types).

experiments with a different architecture. We replace the CRF layer with a linear layer and use token level softmax to predict the tags, keeping the encoder as before. We present the results for four (North Sami, Swedish, German, Galician) of the six languages in Figure 5. Our proposed method CRAL still always outperforms QBC. We observe that only for North Sami does UNS outperform CRAL, which is similar to the results obtained from BRNN/CRF architecture, where the CRAL performs at par with UNS.

5.2 Analysis

In the previous section, we compared the different AL methods by measuring the average POS accuracy. In this section, we perform intrinsic evaluation to compare the quality of the selected data on two aspects:

How similar are the selected and the true data distributions? To measure this similarity, we compare the output tag distribution for each word type in the selected data with the tag distribution in the gold data. This evaluation is necessary because there are a significant number of syncretic word types in the selected data as seen in Table 3. To recap, *syncretic* word types are word types that can have multiple POS tags based on context. We compute the Wasserstein distance (a metric to compute distance between two probability distributions) between the annotated tag distribution and the true tag distribution for each word type z.

$$WD(z) = \sum_{j \in \mathcal{J}_z} p_j^{\text{AL}}(z) - p_j^*(z)$$

where $\mathcal{J}_z$ is the set of output tags for a word type z in the selected active learning data. $p_j^{\text{AL}}(z)$ denotes the proportion of tokens annotated with tag j in the selected data and p_j^* is the proportion of tokens having tag j in the entire gold data. Lower Wasserstein distance suggests high similarity between the selected tag distribution and output tag distribution. Given that each iteration selects unique tokens, this distance can be computed after each of n iterations. Table 4 shows that our proposed strategy CRAL selects data that closely matches the gold data distribution for four out of the six languages.

TARGET LANGUAGE	CRAL	UNS	QBC
German	**0.0465**	0.0801	0.0849
Swedish	**0.0811**	0.1196	0.1013
North Sami	**0.0270**	0.0328	0.0346
Persian	**0.0384**	0.0583	0.0444
Galician	0.0722	0.0953	**0.0674**
Ukrainian	0.0770	0.1067	**0.0665**

Table 4: Wasserstein distance between the output tag distributions of the selected data and the gold data, lower the better. The above results are after 200 annotated tokens, i.e., four AL iterations.

How effective is the AL method in reducing confusion across iterations? Across iterations, as more data is acquired we expect the incorrect predictions from the previous iterations to be rectified in the subsequent iterations, ideally without damaging the accuracy of existing predictions. However, as seen in Table 3, the AL methods have a tendency to select syncretic word types, which suggests that across multiple iterations the same word types could get selected albeit under a different context. This could lead to more confusion, thereby damaging the existing accuracy if the selected type is not a good representative of its annotated tag. Therefore, we calculate the number of existing correct

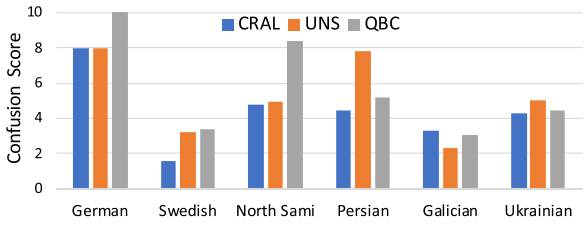

Figure 6: Confusion score measures the percentage of correct predictions in the first iteration which were incorrectly predicted in the second iterations. Lower values suggest that the selected annotations in the subsequent iterations cause less damage on the model trained on the existing annotations.

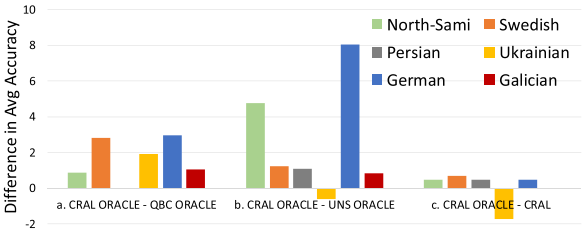

Figure 7: In the *oracle* setting, our method (CRAL-ORACLE) outperforms UNS-ORACLE and QBC-ORACLE in most cases, while the non-oracle CRAL matches the performance of its oracle counterpart. The *y*-axis measures the difference in average accuracy across 20 iterations between the methods being compared.

predictions that were incorrectly predicted in the subsequent iteration, and present the results in Figure 6. A lower value suggests that the AL method was effective in improving overall accuracy without damaging the accuracy from existing annotations, and thereby was successful in reducing confusion. From Figure 6, the proposed strategy CRAL is clearly more effective than the others in most cases in reducing confusion across iterations.

5.3 Oracle Results

In order to check how close to optimal our proposed method CRAL is, we conduct ''oracle'' comparisons, where we have access to the gold labels during data selection. The oracle versions of existing methods UNS-ORACLE and QBC-ORACLE have already been described in Section §2. For our proposed method CRAL, we construct the oracle version as follows:

CRAL-ORACLE: Select the types having the highest incorrect predictions. Within each type, select that output tag that is most incorrectly predicted. This gives the most confusing output tag for a given word type. From the tokens having the most confusing output tag, select the token representative by taking the centroid of their respective contextualized representations, similar to the procedure described in Section § 3.

Figure 7 compares the performance gain of the POS model trained using CRAL-ORACLE over UNS-ORACLE and QBC-ORACLE (Figure 7.a, 7.b). Even under the ''oracle'' setting, our proposed method performs consistently better across all languages (except Ukrainian), unlike the existing methods, as seen in Figure 2. CRAL closely matches the performance of its corresponding ''oracle'' CRAL-ORACLE (Figure 7.c) which suggests that the proposed method is close to an optimal AL method. However, we note that CRAL-ORACLE is not a ''true'' upper bound as for Ukrainian it does not outperform CRAL. We find that for Ukrainian, up to 250 tokens, the oracle method outperforms the non-oracle method after which it underperforms. We hypothesize that this inconsistency is due to noisy annotations in Ukrainian. On analysis we found that the oracle method predicts numerals as NUM but in the gold data some of them are annotated as ADJ. We also find several tokens to have punctuations and numbers mixed with the letters.[7]

In order to verify whether CRAL is accurately selecting data at near-oracle levels, we analyze the intermediate steps leading to the data selection. For each selected word type $z \in X_{\text{LABEL}}$, we analyze how well our proposed method of weighting encoder representations with the model confidence of the most confused tag and taking the centroid actually succeeds at ''representative'' token selection. If this is indeed the case, tokens in the vicinity of the centroid should also have the same ''most confused tag'' as their predicted label and thereby be mis-classfied instances. To verify this hypothesis we compare how many of the 100 tokens closest to the centroid (in the representation space) ($X_{\text{NN}}(z)$) are truly misclassified. This score is given by $p(z)$ for each selected word-type z:

$$X_{\text{NN}}(z) = \text{b-}\underset{x_{i,t}=z \in D}{\arg\min} |\mathbf{c}_{i,t} - \mathbf{c}_z|$$

$$p(z) = \frac{|\hat{y}_{i,t} \neq y^*_{i,t}|}{|X_{\text{NN}}(z)|}$$

[7]This is also noted in the UD page: https://universaldependencies.org/treebanks/uk_iu/index.html.

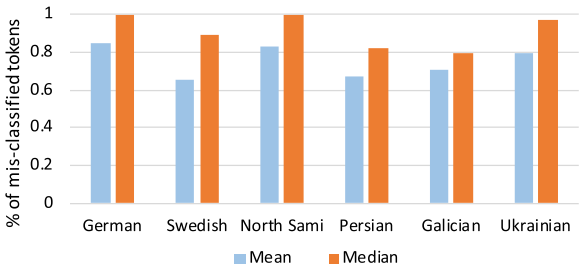

Figure 8: We report the mean and median of $\mathbf{p}$ over all the 50 token-tag pairs selected by the first AL iteration of CRAL. We see that across all languages majority of the token-tag pairs satisfy the criteria of using weighted representations with centroid for token selection.

where $b = 100$. $\mathbf{c}_z$ is the contextualized representation of the representative instance for the word-type z, namely, the centroid and $\mathbf{c}_{i,t}$ is the contextualized representation of z's token instance $x_{i,t}$. $y^*_{i,t}$ and $\hat{y}_{i,t}$ are the true and predicted labels of $x_{i,t}$. We report the average and median of $\mathbf{p}$ across all the selected tokens of the first AL iteration in Figure 8. We see that for all languages the median is high (i.e., > 0.8) which suggests that the majority of the token-tag pairs satisfy this criteria, thus supporting the step of weighting the token representations and choosing the centroid for annotation.

We also compare the percent of token-tag overlap between the data selected from CRAL with its oracle counterpart: CRAL-ORACLE. For the first AL iteration, the proposed method CRAL has more than 50% overlap with the oracle method for all languages, providing some evidence as to why CRAL is matching the oracle performance.

5.4 Effect of Cross-View Training

As mentioned in Section § 4.2, we use CVT to not only improve our model overall but also to have a well-calibrated model that can be important for active learning. A model is well-calibrated when a model's predicted probabilities over the outcomes reflects the true probabilities over these outcomes (Nixon et al., 2019). We use Static Calibration Error (SCE), a metric proposed by Nixon et al. (2019) to measure the model calibration. SCE bins the model predictions separately for each output tag probability and computes the calibration error within each bin which is averaged across all the bins to produce a single score. For each output tag, bins are created by sorting the predictions based on the output class probability. Hence, the first 10% are placed in bin 1, the next 10% in bin 2, and so on. We conduct two ablation experiments to measure the effect of CVT. First, we train a joint POS model on English and Norwegian datasets using all available training data, and evaluate on the English test set. Second, we use this pre-trained model and fine-tune on 200 randomly sampled German data and evaluate on German test data. We train models with and without CVT, denoted by +/- in Table 5. We find that with CVT results both in higher accuracy as well as lower calibration error (SCE). This effect of CVT is much more pronounced in the second experiment, which presents a low-resource scenario and is common in an AL framework.

EXPERIMENT SETTING	CVT	SCE	ACCURACY
EN + NO → EN	−	0.0190	95.53
	+	**0.0174**	**95.58**
EN + NO + DE-200 → DE	−	0.1658	69.90
	+	**0.1391**	**74.61**

Table 5: Evaluating the effect of CVT across two experimental settings. EN: English, NO: Norwegian, DE-200: 200 German annotations. Left of ''→'' are the pre-training languages and the right of ''→'' is the language on which this model is evaluated. Accuracy measures the POS model performance (higher is better) and SCE measures the model calibration (lower is better).

6 Human Annotation Experiment

In this section, we apply our proposed approach on Griko, an endangered language spoken by around 20,000 people in southern Italy, in the Grecìa Salentina area southeast of Lecce. The only available online Griko corpus, referred to as UoI (Lekakou et al., 2013),[8] consists of 330 utterances by nine native speakers having POS annotations. Additionally, Anastasopoulos et al. (2018) collected, processed, and released 114 stories, of which only the first 10 stories were annotated by experts and have gold-standard annotations. We conduct human annotation experiments on the remaining unannotated stories in order to compare the different active learning methods.

Setup: We use Modern Greek and Italian as the two related languages to train our initial

[8] http://griko.project.uoi.gr.

	AL	ITERATION-0	ITERATION-1	ITERATION-2	ITERATION-3	IA Agr.	WD
Linguist-1	CRAL	52.93	**63.42** (10)	**69.07** (10)	65.16 (16)	0.58	0.281
	QBC	52.93	55.82 (15)	62.03 (17)	**66.51** (15)	0.68	0.243
	UNS	52.93	56.14 (15)	57.04 (15)	65.73 (11)	0.58	0.379
Linguist-2	CRAL	52.93	**61.24** (15)	**67.24** (20)	**67.05** (18)	0.70	0.346
	QBC	52.93	56.52 (20)	65.96 (20)	66.71 (17)	0.72	0.245
	UNS	52.93	55.45 (17)	58.80 (17)	65.73 (20)	0.70	0.363
Linguist-3 (Expert)	CRAL	52.93	**65.63**	**69.17**	**68.09**	–	0.159
	QBC	52.93	60.50	65.69	56.20	–	0.170
	UNS	52.93	58.51	64.26	65.93	–	0.125

Table 6: Griko test set POS accuracy after each AL annotation iteration. Each iteration consists of 50 token-level annotations. The number in parentheses is the time in minutes required for annotation. The IA AGR. column reports the inter-annotator agreement against the expert linguist for the first iteration. WD is the Wasserstein distance between the selected tokens and the test distribution.

POS model.[9] To further improve the model, we fine-tune on the UoI corpus, which consists of 360 labeled sentences. We evaluate the AL performance on the 10 gold-labelled stories from Anastasopoulos et al. (2018), of which the first two stories, comprising 143 labeled sentences, are used as the validation set and the remaining 800 labeled sentences form the test set. We use the unannotated stories as our unlabeled pool. We compare CRAL with UNS and QBC, conducting three AL iterations for each method, where each iteration selects roughly 50 tokens for annotation. The annotations are provided by two linguists familiar with modern Greek and somewhat familiar with Griko. To familiarize the linguists with the annotation interface, a practice session was conducted in modern Greek. In the interface, tokens that need to be annotated are highlighted and presented with their surrounding context. The linguist then simply selects the appropriate POS tag for each highlighted token. Because we do not have gold annotations for these experiments, we also obtained annotations from a third linguist who is more familiar with Griko grammar.

Results: Table 6 presents the results on three iterations for each AL method, with our proposed method CRAL outperforming the other methods in most cases. We note that we found several frequent tokens (i.e., 863/13,740 tokens) in the supposedly gold-standard Griko test data to be inconsistently annotated. Specifically, the original annotations did not distinguish between coordinating (`CCONJ`) and subordinating conjunctions (`SCONJ`), unlike the UD schema. As a result, when converting the test data to the UD schema all conjunctions were tagged as subordinating ones. Our annotation tool, however, allowed for either `CCONJ` or `SCONJ` as tags and the annotators did make use of them. With the help of a senior Griko linguist (Linguist-3), we identified a few types of conjunctions that are always coordinating: variations of ''and'' (`ce` and `c'`), and of ''or'' (`e` or `i`). We fixed these annotations and used them in our experiments.

For Linguist-1, we observe a decrease in performance in Iteration-3. One possible reason for this decrease is attributed to Linguist-1's poor annotation quality, which is also reflected in their low inter-annotator agreement scores. We observe a slight decrease for other linguists, which we hypothesize is due to domain mismatch between the annotated data and the test data. In fact, the test set stories and the unlabeled ones originate from different time periods spanning a century, which can lead to slight differences in orthography and usage. For instance, after three AL iterations, the token ''i'' had been annotated as CONJ twice and DET once, whereas in the test data all instances of ''i'' are annotated as DET. Similar to the simulation experiments, we compute the confusion score for all linguists in Figure 9. We find that, unlike in the simulation experiments, a model trained with UNS causes less damage on the existing annotations as compared to CRAL. However, we note that the model performance from the UNS annotations is much lower than CRAL to begin with.

[9]With Italian being the dominant language in the region, code switching phenomena appear in the Griko corpora.

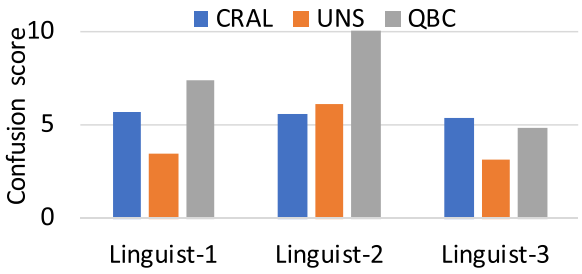

Figure 9: Confusion scores for the three Griko linguists. Lower values suggest that the selected annotations in the subsequent iterations cause less damage on the model trained on existing annotation.

We also compute the inter-annotator agreement at Iteration-1 with the expert (Linguist-3) (Table 6). We find that the agreement scores are lower than one would expect (c.f. the annotation test run on modern Greek, for which we have gold annotations, yielded much higher interannotator agreement scores over 90%). The justification probably lies with our annotators having limited knowledge of Griko grammar, while our AL methods require annotations for ambiguous and ''hard'' tokens. However, this is a common scenario in language documentation where often linguists are required to annotate in a language they are not very familiar with, which makes this task even more challenging. We also recorded the annotation time needed by each linguist for each iteration in Table 6. Compared with the UNS method, the linguists annotated (avg.) 2.5 minutes faster using our proposed method which suggests that UNS tends to select harder data instances for annotation.

Similar to the simulation experiments, we report the Wasserstein distance (WD) for all linguists in Table 6. However, unlike in the simulation setting where the WD was computed with the gold training data, for the human experiments we do not have access to the gold annotations and therefore computed WD with the gold test data which however, is from a slightly different domain, which affects the results somewhat. We observe that QBC has lower WD scores for Linguist-1 and Linguist-2 and UNS for Linguist-3. On further analysis, we find that even though QBC has lower WD, it also has the least coverage of the test data—that is, it has the fewest number of annotated tokens which are present in the test data, as shown in Table 7. We would like to note that a lower WD score does not necessarily translate to better tagging accuracy because the WD metric is only informing us how good an AL strategy is in selecting data that matches closely the gold output tag distribution for that selected data.

	CRAL	UNS	QBC
Linguist-1	90	95	72
Linguist-2	84	88	72
Linguist-3	74	90	61

Table 7: Each cell denotes the number of annotated tokens that are also present in the test data.

7 Related Work

Active Learning for POS Tagging: AL has been widely used for POS tagging. Garrette and Baldridge (2013) use a graph-based label propagation to generalize initial POS annotations to the unlabeled corpus. Further, they find that under a constrained time setting, type-level annotations prove to be more useful than token-level annotations. In line with this, Fang and Cohn (2017) also select informative word types based on uncertainty sampling for low-resource POS tagging. They also construct a tag dictionary from these type-level annotations and then propagate the labels across the entire unlabeled corpus. However, in our initial analysis on uncertainty sampling, we found that adding label-propagation harmed the accuracy in certain languages because of prevalent syncretism. Ringger et al. (2007) present different variations of uncertainty-sampling and QBC methods for POS tagging. Similar to Fang and Cohn (2017), they find uncertainty sampling with frequency bias to be the best strategy. Settles and Craven (2008) present a nice survey on the different active learning strategies for sequence labeling tasks, and Marcheggiani and Artières (2014) discuss the strategies for acquiring partially labeled data. Sener and Savarese (2018) propose a *core-set* selection strategy aimed at finding the subset that is competitive across the unlabeled dataset. This work is most similar to ours with respect to using geometric center points as being the most representative. However, to the best of our knowledge, none of the existing works are targeted at reducing confusion within the output classes.

Low-resource POS Tagging: Several cross-lingual transfer techniques have been used for improving low-resource POS tagging. Cotterell and Heigold (2017) and Malaviya et al. (2018) train a joint neural model on related high-resource languages and find it be very effective on low-resource languages. The main advantage of these methods is that they do not require any parallel text or dictionaries. Das and Petrov (2011), Täckström et al. (2013), Yarowsky et al. (2001), and Nicolai and Yarowsky (2019) use annotation projection methods to project POS annotations from one language to another. However, annotation projection methods use parallel text, which often might not be of good quality for low-resource languages.

8 Conclusion

We have presented a novel active learning method for low-resource POS tagging that works by reducing confusion between output tags. Using simulation experiments across six typologically diverse languages, we show that our confusion-reducing strategy achieves higher accuracy than existing methods. Further, we test our approach under a true setting of active learning where we ask linguists to document POS information for an endangered language, Griko. Despite being unfamiliar with the language, our proposed method achieves performance gains over the other methods in most iterations. For our next steps, we plan to explore the possibility of adapting our proposed method for *complete* morphological analysis, which poses an even harder challenge for AL data selection due to the complexity of the task.

Acknowledgments

The authors are grateful to the anonymous reviewers and the Action Editor who took the time to provide many interesting comments that made the paper significantly better, and to Eleni Antonakaki and Irini Amanaki, for participating in the human annotation experiments. This work is sponsored by the Dr. Robert Sansom Fellowship, the Waibel Presidential Fellowship, and the National Science Foundation under grant 1761548.

References

Antonios Anastasopoulos, Marika Lekakou, Josep Quer, Eleni Zimianiti, Justin DeBenedetto, and David Chiang. 2018. Part-of-speech tagging on an endangered language: A parallel griko-italian resource. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2529–2539, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ankita and K. A. Abdul Nazeer. 2018. Part-of-Speech Tagging and Named Entity Recognition using Improved Hidden Markov Model and Bloom Filter. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 1072–1077. IEEE.

Kedar Bellare and Andrew McCallum. 2007. Learning extractors from unlabeled text using relevant databases. In *Sixth International Workshop on Information Integration on the Web*.

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic Tagging with a meta-BiLSTM Model over Context Sensitive Token Encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia. Association for Computational Linguistics, **DOI:** https://www.aclweb.org/anthology/P18-1246

Aditi Chaudhary, Jiateng Xie, Zaid Sheikh, Graham Neubig, and Jaime Carbonell. 2019. A Little Annotation does a Lot of Good: A Study in Bootstrapping Low-resource Named Entity Recognizers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5164–5174, Hong Kong, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D19-1520

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-Supervised

Sequence Modeling with Cross-View Training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Ryan Cotterell and Georg Heigold. 2017. Cross-Lingual Character-Level Neural Morphological Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D17-1078

Dipanjan Das and Slav Petrov. 2011. Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 600–609, Portland, Oregon, USA. Association for Computational Linguistics.

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

Meng Fang and Trevor Cohn. 2017. Model Transfer for Tagging Low-Resource Languages using a Bilingual Dictionary. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 587–593, Vancouver, Canada. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P17-2093

Dan Garrette and Jason Baldridge. 2013. Learning a Part-of-Speech Tagger from Two Hours of Annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, Atlanta, Georgia. Association for Computational Linguistics.

Harald Hammarström, Robert Forkel, and Martin Haspelmath. 2018. Glottolog 3.3. Max Planck Institute for the Science of Human History. Jena.

Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. 2010. Active Learning by Querying Informative and Representative Examples. In *Advances in Neural Information Processing Systems*, pages 892–900.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for Sequence Tagging. *arXiv preprint arXiv:1508.01991*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, and Greg Corrado. 2017. Googles Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351. **DOI:** https://doi.org/10.1162/tacl_a_00065

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal Morphology. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, CA. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/N16-1030

Marika Lekakou, Valeria Baldissera, and Antonios Anastasopoulos. 2013. Documentation and

analysis of an endangered language: aspects of the grammar of griko.

David D. Lewis. 1995. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–254. **DOI:** https://doi.org/10.1145/215206.215366

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing Transfer Languages for Cross-Lingual Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074. Association for Computational Linguistics,

Chaitanya Malaviya, Matthew R. Gormley, and Graham Neubig. 2018. Neural Factor Graph Models for Cross-Lingual Morphological Tagging. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2653–2663, Melbourne, Australia. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P18-1247

Diego Marcheggiani and Thierry Artières. 2014. An experimental comparison of active learning strategies for partially labeled sequences. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 898–906, Doha, Qatar. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/D14-1097

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101. **DOI:** https://doi.org/10.18653/v1/W18-6011

Garrett Nicolai and David Yarowsky. 2019. Learning Morphosyntactic Analyzers from the Bible via Iterative Annotation Projection across 26 Languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1765–1774, Florence, Italy. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P19-1172

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, and Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Jeremy Nixon, Mike Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring calibration in Deep Learning. *arXiv preprint arXiv:1904.01685*.

Eric Ringger, Peter McClanahan, Robbie Haertel, George Busby, Marc Carmen, James Carroll, Kevin Seppi, and Deryle Lonsdale. 2007. Active Learning for part-of-speech Tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 101–108. **DOI:** https://doi.org/10.3115/1642059.1642075

Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*.

Burr Settles. 2009, Active Learning literature survey, University of Wisconsin-Madison Department of Computer Sciences.

Burr Settles and Mark Craven. 2008. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural*

Language Processing, pages 1070–1079, Honolulu, HI. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1613715.1613855

Aditya Siddhant, Ankur Bapna, Henry Tsai, Jason Riesa, Karthik Raman, Melvin Johnson, Naveen Ari, and Orhan Firat. 2020. In *Evaluating the Cross-lingual Effectiveness of Massively Multilingual Neural Machine Translation*. **DOI:** https://doi.org/10.1609/aaai.v34i05.6414

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Tagging. 2013. Token and type constraints for cross-lingual part-of-speech Tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12. **DOI:** https://doi.org/10.1162/tacl_a_00205

Nivre Joakim, Blokland Rogier, Partanen Niko, Rießler Michael, and Rueter Jack. 2018. Universal Dependencies 2.3.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Zhe Wang, Xiaoyi Liu, Limin Wang, Yu Qiao, Xiaohui Xie, and Charless Fowlkes. 2018. Structured Triplet Learning with POS-tag Guided Attention for Visual Question Answering. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1888–1896, IEEE. **DOI:** https://doi.org/10.1109/WACV.2018.00209

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–8. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1072133.1072187

Imed Zitouni and Radu Florian. 2008. Mention detection crossing the language barrier. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 600–609. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1613715.1613789