Interactive Refinement of Cross-Lingual Word Embeddings

Michelle Yuan*
University of Maryland
myuan@cs.umd.edu

Mozhi Zhang*
University of Maryland
mozhi@cs.umd.edu

Benjamin Van Durme
Johns Hopkins University
vandurme@jhu.edu

Leah Findlater
University of Washington
leahkf@uw.edu

Jordan Boyd-Graber
University of Maryland
jbg@umiacs.umd.edu

arXiv:1911.03070v3 [cs.CL] 7 Oct 2020

Abstract

Cross-lingual word embeddings transfer knowledge between languages: models trained on high-resource languages can predict in low-resource languages. We introduce CLIME, an interactive system to quickly refine cross-lingual word embeddings for a given classification problem. First, CLIME ranks words by their salience to the downstream task. Then, users mark similarity between keywords and their nearest neighbors in the embedding space. Finally, CLIME updates the embeddings using the annotations. We evaluate CLIME on identifying health-related text in four low-resource languages: Ilocano, Sinhalese, Tigrinya, and Uyghur. Embeddings refined by CLIME capture more *nuanced* word semantics and have higher test accuracy than the original embeddings. CLIME often improves accuracy faster than an active learning baseline and can be easily combined with active learning to improve results.

1 Introduction

Modern text classification requires large labeled datasets and pre-trained word embeddings (Kim, 2014; Iyyer et al., 2015; Joulin et al., 2017). However, scarcity of both labeled and unlabeled data holds back applications in low-resource languages. Cross-lingual word embeddings (Mikolov et al., 2013a, CLWE) can bridge the gap by mapping words from different languages to a shared vector space. Using CLWE features, models trained in a resource-rich language (e.g., English) can predict labels for other languages.

The success of CLWE relies on the domain and quality of training data (Søgaard et al., 2018). While these methods have impressive word translation accuracy, they are not tailored for downstream tasks such as text classification (Glavas et al., 2019; Zhang et al., 2020a). We develop **CL**assifying **I**nteractively with **M**ultilingual **E**mbeddings (CLIME), that efficiently specializes CLWE with *human interaction*.[1] Given a pre-trained CLWE, a bilingual speaker in the loop reviews the nearest-neighbor words. CLIME capitalizes on the intuition that neighboring words in an ideal embedding space should have similar semantic attributes.

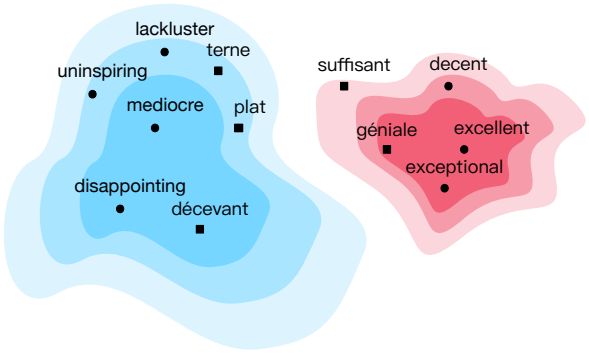


Figure 1: A hypothetical topographic map of an English–French embedding space tailored for sentiment analysis. Dots are English words, and squares are French words. Positive sentiment words are grouped in a clime (red), while negative sentiment words are grouped in another clime (blue). These climes help sentiment analysis.

In an analogy to geographic *climes*—zones with distinctive meteorological features—we call areas in the embedding space where words share similar semantic features climes. Our goal is to convert neighborhoods in the embedding space into *classification climes* with words that induce similar labels for a given classification task. For example, in the embedding for English–French sentiment analysis, positive sentiment words such as "excellent", "exceptional", and their French translations are together, while "disappointing", "lackluster", and their translations cluster together elsewhere (Fig-

* indicates equal contribution

[1] https://github.com/forest-snow/clime-ui

ure 1). Curating words in the embedding space and refining climes should help downstream classifiers.

First, CLIME uses loss gradients in downstream tasks to find keywords with high salience (Section 2.1). Focusing on these keywords allows the user to most efficiently refine CLWE by marking their similarity or dissimilarity (Section 2.2). After collecting annotations, CLIME pulls similar words closer and pushes dissimilar words apart (Section 3), establishing desired climes (Figure 1).

Quickly deploying cross-lingual NLP systems is particularly important in global public health emergencies, so we evaluate CLIME on a cross-lingual document classification task for four low-resource languages: Ilocano, Sinhalese, Tigrinya, and Uyghur (Section 4). CLIME is effective in this low-resource setting because a bilingual speaker can significantly increase test accuracy on identifying health-related documents in less than an hour.

CLIME is related to active learning (Settles, 2009), which also improves a classifier through user interaction. Therefore, we compare CLIME with an active learning baseline that asks a user to label target language documents. Under the same annotation time constraint, CLIME often has higher accuracy. Furthermore, the two methods are complementary. Combining active learning with CLIME increases accuracy even more, and the user-adapted model is competitive with a large, resource-hungry multilingual transformer (Conneau et al., 2020).

2 Interactive Neighborhood Reshaping

This section introduces the interface designed to solicit human feedback on neighborhoods of CLWE and our keyword selection criterion. Suppose that we have two languages with vocabulary $\mathcal{V}_1$ and $\mathcal{V}_2$. Let $\mathbf{E}$ be a pre-trained CLWE matrix, where $\mathbf{E}_w$ is the vector representation of word type w in the joint vocabulary $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$. Our goal is to help a bilingual novice (i.e., not a machine learning expert) improve the CLWE $\mathbf{E}$ for a downstream task through inspection of neighboring words.

2.1 Keyword Selection

With limited annotation time, users cannot vet the entire vocabulary. Instead, we need to find a small salient subset of *keywords* $\mathcal{K} \subseteq \mathcal{V}$ whose embeddings, if vetted, would most improve a downstream task. For example, if the downstream task is sentiment analysis, our keywords set should include sentiment words such as "good" and "bad". Prior work in active learning solicits keywords using information gain (Raghavan et al., 2006; Druck et al., 2009; Settles, 2011), but this cannot be applied to continuous embeddings. Li et al. (2016) suggest that the contribution of one dimension of a word embedding to the loss function can be approximated by the absolute value of its partial derivative, and therefore they use partial derivatives to visualize the behavior of neural models. However, rather than understanding the importance of individual dimensions, we want to compute the salience of an *entire word vector*. Therefore, we extend their idea by defining the salience of a word embedding as the *magnitude* of the loss function's gradient. This score summarizes salience of all dimensions from a word embedding. Formally, let $\mathbf{x} = \langle x_1, x_2, \cdots, x_n \rangle$ be a document of n words with label y; let L be the training loss function of the downstream model. We measure the example-level salience of word x_i in document $\mathbf{x}$ as

$$S_{\mathbf{x}}(x_i) = \left\| \nabla_{\mathbf{E}_{x_i}} L(\mathbf{x}, y) \right\|_2 . \quad (1)$$

Equation 1 measures the local contribution of a token in one document, but we are interested in the global importance of a word type across many documents. To compute the global salience score of a word type w, we add example-level salience scores of all token occurrences of a word type w in a large labeled dataset $\mathbf{X}$ and multiply by the inverse document frequency (IDF) of w:

$$S(w) = \text{IDF}(w, \mathbf{X}) \cdot \sum_{\mathbf{x} \in \mathbf{X}: w \in \mathbf{x}} S_{\mathbf{x}}(w). \quad (2)$$

The IDF term is necessary because it discounts *stop words* with high document frequency (e.g., "the" and "of"). These words are often irrelevant to the downstream task and thus have low example-level salience, but they have high total salience because they appear in many examples.

Based on Equation 2, we choose the top-s most salient words as the keyword set $\mathcal{K}$. The hyper-parameter s is the number of keywords displayed to the user, which controls the length of a CLIME session. We limit s to fifty in experiments.

2.2 User Interaction

For each keyword k, we want to collect a positive set $\mathcal{P}_k$ with semantically similar words, and a negative set $\mathcal{N}_k$ with unrelated words. To specialize embeddings for a classification task, we ask the user to consider semantic similarity as *inducing a*

Figure 2: The CLIME interface displays a keyword on top while its nearest neighbors in the two languages appear in the two columns below. A user can accept or reject each neighbor, and add new neighbors by typing them in the "add word" textboxes. They may also click on any word to read its context in the training set.

similar label. For example, if the task is English–French sentiment analysis, then "good" should be considered similar to "excellent" and "génial" but dissimilar to "bad" and "décevant". On the interface, the keyword k is displayed on the top, and its nearest neighbors in the two languages are arranged in two columns (Figure 2). The neighbors are the words w with embeddings $\mathbf{E}_w$ closest to $\mathbf{E}_k$ in cosine similarity. The number of displayed nearest neighbors can be adjusted as a hyperparameter, which also controls the session length. For each nearest neighbor, the user can either: (1) press on the green checkmark to add a positive neighbor to $\mathcal{P}_k$, (2) press on the red "X" mark to add a negative neighbor to $\mathcal{N}_k$, or (3) leave an uncertain neighbor alone. The "add word" textbox lets the user add words that are not in the current neighbor list. The added word can then be marked as positive or negative. Section 3 explains how CLIME refines the embeddings with the feedback sets $\mathcal{P}$ and $\mathcal{N}$. The interface also provides a word concordance—a brief overview of the contexts where a word appears—to disambiguate and clarify words. Users can click on any word to find example sentences.

3 Fitting Word Embeddings to Feedback

After receiving user annotations, CLIME updates the embeddings to reflect their feedback. The algorithm reshapes the neighborhood so that words near a keyword share similar semantic attributes. Together, these embeddings form desired task-specific connections between words across languages. Our update equations are inspired by ATTRACT-REPEL (Mrkšić et al., 2017), which fine-tunes word embeddings with synonym and antonym constraints. The objective in ATTRACT-REPEL pulls synonyms closer to and pushes antonyms further away from their nearest neighbors. This objective is useful for large lexical resources like BabelNet (Navigli and Ponzetto, 2010) with hundreds of thousands linguistic constraints, but our pilot experiment suggests that the method is not suitable for smaller constraint sets. Since CLIME is designed for low-resource languages, we optimize an objective that reshapes the neighborhood more drastically than ATTRACT-REPEL.

3.1 Feedback Cost

For each keyword $k \in \mathcal{K}$, we collect a positive set $\mathcal{P}_k$ and a negative set $\mathcal{N}_k$ (Section 2.2). To refine

embeddings $\mathbf{E}$ with human feedback, we increase the similarity between k and each positive word $p \in \mathcal{P}_k$, and decrease the similarity between k and each negative word $n \in \mathcal{N}_k$. Formally, we update the embeddings $\mathbf{E}$ to minimize the following:

$$C_f(\mathbf{E}) = \sum_{k \in \mathcal{K}} \left(\sum_{n \in \mathcal{N}_k} \mathbf{E}_k^\top \mathbf{E}_n - \sum_{p \in \mathcal{P}_k} \mathbf{E}_k^\top \mathbf{E}_p \right), \tag{3}$$

where $\mathbf{E}_k^\top \mathbf{E}_n$ measures the similarity between the keyword k and a negative word n, and $\mathbf{E}_k^\top \mathbf{E}_p$ measures the similarity between the keyword k and a positive word p. Minimizing C_f is equivalent to maximizing similarities of positive pairs while minimizing similarities of negative pairs.

3.2 Topology-Preserving Regularization

Prior embedding post-processing methods emphasize regularization to maintain the topology—or properties that should be preserved under transformations—of the embedding space (Mrkšić et al., 2016; Mrkšić et al., 2017; Glavaš and Vulić, 2018). If the original CLWE brings certain translations together, those translated words should remain close after updating the embeddings. The topology also encodes important semantic information that should not be discarded. Therefore, we also include the following regularization term:

$$R(\mathbf{E}) = \sum_{w \in \mathcal{V}} \left\| \hat{\mathbf{E}}_w - \mathbf{E}_w \right\|_2^2. \tag{4}$$

Minimizing $R(\mathbf{E})$ prevents $\mathbf{E}$ from drifting too far away from the original embeddings $\hat{\mathbf{E}}$.

The final cost function combines the feedback cost (Equation 3) and the regularizer (Equation 4):

$$C(\mathbf{E}) = C_f(\mathbf{E}) + \lambda R(\mathbf{E}), \tag{5}$$

where the hyperparameter λ controls the strength of the regularizer. The updated embeddings enforce constraints from user feedback while preserving other structures from the original embeddings. After tuning in a pilot user study, we set λ to one. We use the Adam optimizer (Kingma and Ba, 2015) with default hyperparameters.

4 Cross-Lingual Classification Experiments

We evaluate CLIME on cross-lingual document-classification (Klementiev et al., 2012), where we build a text classifier for a low-resource target language using labeled data in a high-resource source language through CLWE. Our task identifies whether a document describes a medical emergency, useful for planning disaster relief (Strassel and Tracey, 2016). The source language is English and the four low-resource target languages are Ilocano, Sinhalese, Tigrinya, and Uyghur.

Ilocano	... Nagtalinaed dagiti pito a balod ti Bureau of Jail Management and Penology (BJMP) ditoy ciudad ti Laoag iti isolation room gapo iti tuko ...
English	... Seven inmates from the Bureau of Jail Management and Penology (BJMP), Laoag City, have been transferred to the isolation room due to chicken pox ...

Table 1: Excerpt of a positive Ilocano test example (top) and its English translation (bottom) that describes a medical emergency.

Our experiments confirm that a bilingual user can quickly improve the test accuracy of cross-lingual models through CLIME. Alternatively, we can ask an annotator to improve the model by labeling more training documents in the target language. Therefore, we compare CLIME to an active learning baseline that queries the user for document labels; CLIME often improves accuracy faster. Then, we combine CLIME and active learning to show an even faster improvement of test accuracy.

Comparing active learning to CLIME may seem unfair at first glance. In theory, document labeling only requires target language knowledge, while CLIME learns from a bilingual user. In practice, researchers who speak a high-resource language provide instructions to the annotator and answer their questions, so bilingual knowledge is usually required in document labeling for low-resource languages. Moreover, CLIME is complementary to active learning, as combining them gives the highest accuracy across languages.

We also experiment with refining the same set of keywords with multiple rounds of user interaction. The repeated sessions slightly improve test accuracy on average. Finally, we compare with XLM-R (Conneau et al., 2020), a state-of-the-art multilingual transformer. Despite using fewer resources, CLIME has competitive results.

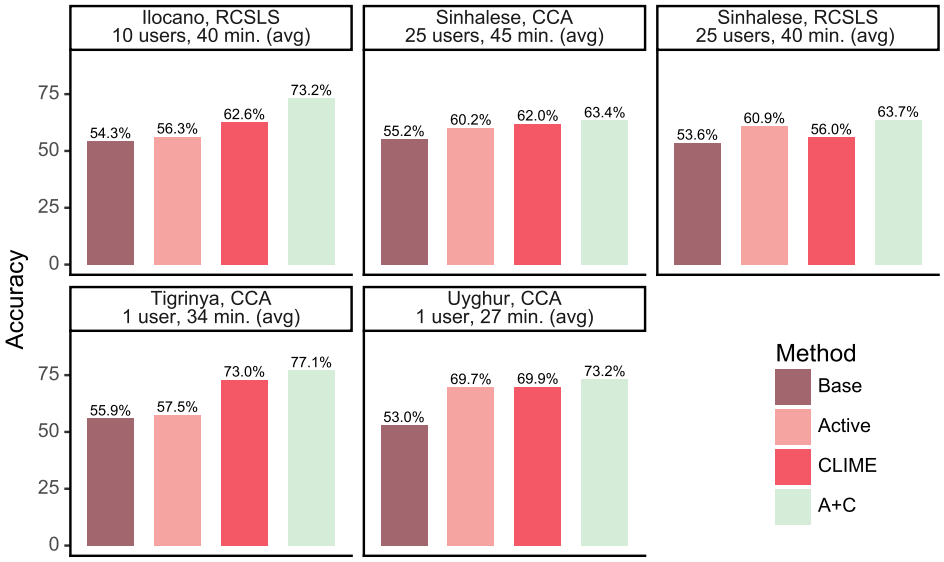


Figure 3: Test accuracy of four methods on four target languages and two CLWE methods. **Base** uses the original CLWE and the original training set. **Active** uses the original CLWE and a training set augmented by active learning. We select and label fifty *target language* documents by uncertainty sampling and combine them with the source language training set. **CLIME** uses the CLWE refined by CLIME and the original training set. **A+C** uses the CLWE refined by CLIME and a training set augmented by active learning. We control the number of user interactions so that **Active**, **CLIME**, and **A+C** require the similar interaction time (Section 4.2). The Sinhalese and Ilocano results are averaged over multiple users, while we only have one user for other languages. Each subcaption indicates the target language, embedding alignment, number of users, and average time per user. **CLIME** has higher accuracy than **Active** on four of the five embeddings, and the combined **A+C** model has the highest.

4.1 Experiment Setup

Labeled Data. We train models on 572 English documents and test on 48 Ilocano documents, 58 Sinhalese documents, 158 Tigrinya documents, and 94 Uyghur documents. The documents are extracted from LORELEI language packs (Strassel and Tracey, 2016), a multilingual collection of documents of emergencies with a public health component.[2] To simplify the task, we consider a binary classification problem of detecting whether the documents are associated with medical needs. Table 1 shows an example document. To balance the label distribution, we sample an equal number of negative examples.

Word Embeddings. To transfer knowledge between languages, we build CLWE between English and each target language. We experiment with two methods to pre-train CLWE: (1) train monolingual embeddings with word2vec (Mikolov et al., 2013b) and align with CCA (Faruqui et al., 2015; Ammar et al., 2016), (2) train monolingual embeddings with fastText (Bojanowski et al., 2017) and align with RCSLS (Joulin et al., 2018). The English embeddings are trained on Wikipedia and the target language embeddings are trained on unlabeled documents from the LORELEI language packs. For alignment, we use the small English dictionary in each pack. Low-resource language speakers are hard to find, so we do not try all combinations of languages and CLWE: we use CCA embeddings for Tigrinya and Uyghur, RCSLS embeddings for Ilocano. Since Sinhalese speakers are easier to find, we experiment with both CLWE for Sinhalese.

Text Classifier. Our classifier is a convolutional neural network (Kim, 2014). Each document is represented as the concatenation of word embeddings and passed through a convolutional layer, followed by max-pooling and a final softmax layer. To preserve cross-lingual alignments, we freeze embeddings during training. This simple model is effective in low-resource cross-lingual settings (Chen et al., 2018; Schwenk and Li, 2018). We minimize cross-entropy on the training set by running Adam (Kingma and Ba, 2015) with default hyperparameters for thirty epochs. All experiments use GeForce GTX 1080 GPU and 2.6 GHz AMD Opteron 4180 processor.

[2]Download from https://www.ldc.upenn.edu

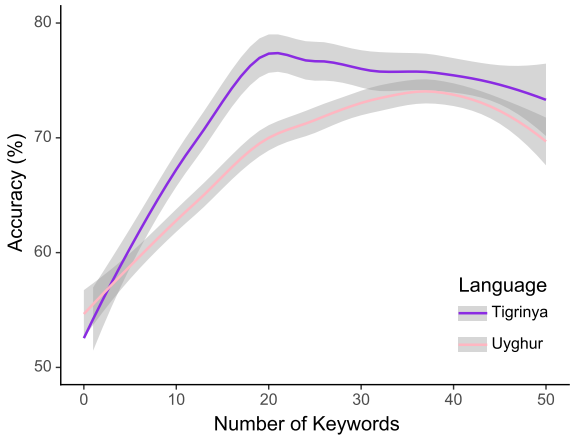


Figure 4: For Uyghur (pink) and Tigrinya (purple), we compare test accuracy between sets of CLWE that differ in the number of keywords used to refine them. The leftmost point corresponds to the **Base** model in Figure 3, while the rightmost point corresponds to the **CLIME** model. Test accuracy generally improves with more feedback at the beginning but slightly drops after reaching an optimal number of keywords.

User Study. We use Upwork to hire participants who are fluent in both English and the target language.[3] Low-resource language speakers are hard to find, so we have a different number of users for each language. We hire ten Ilocano users and twenty-five Sinhalese users. For additional case studies, we hire one Tigrinya user and one Uyghur user. Each user annotates the fifty most salient keywords, which takes less than an hour (Figure 3). For each keyword, we show five nearest neighbors for each language. Each user provides about nine constraints for each keyword.

4.2 Comparisons

After receiving feedback, we update the embeddings (Section 3). We evaluate the new embeddings by retraining a classifier. For each set of embeddings, we train ten models with different random seeds and report average test accuracy.

We compare a classifier trained on the updated embeddings (**CLIME** in Figure 3) against two baselines. The first baseline is a classifier trained on original embeddings (**Base** in Figure 3). If we have access to a bilingual speaker, an alternative to using CLIME is to annotate more training documents in the target language. Therefore, we also compare CLIME to uncertainty sampling (Lewis and Gale, 1994), an active learning method that asks a user to label documents (**Active** in Figure 3). We choose a set of fifty documents where model outputs have the highest entropy from a set of unlabeled *target language* documents and ask an annotator to label them as additional training documents. We then retrain a model on both the English training set and the fifty target language documents, using the original embeddings. For each model, a human annotator labels fifty documents within forty to fifty minutes. This can either be slower or take approximately the same time as an average CLIME session (Figure 3). Thus, any improvements in accuracy using CLIME are even more impressive given that **Active** is no faster than CLIME.

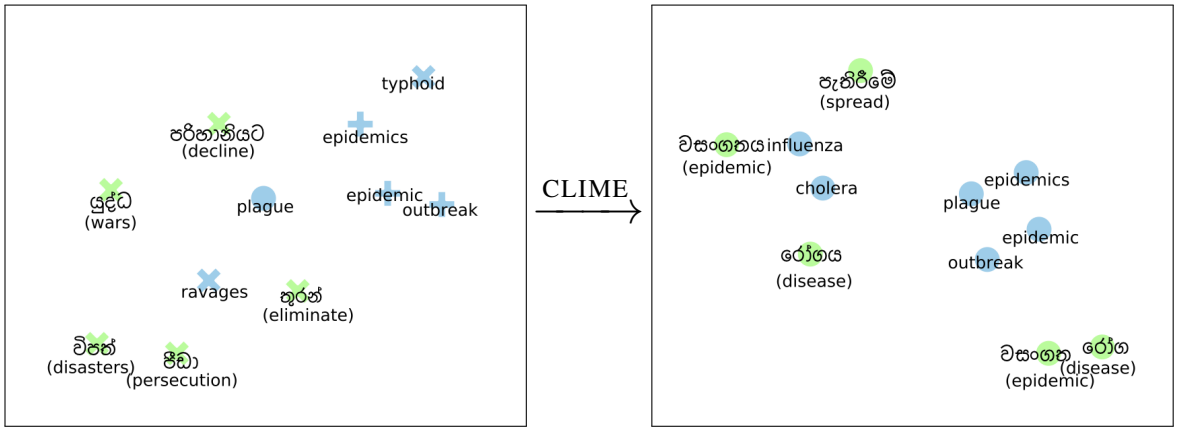
Finally, we explore combining active learning and CLIME (**A+C** in Figure 3). Document-level and word-level interactions are complementary, so using both may lead to higher accuracy. To keep the results comparable, we allocate half of the user interaction time to active learning, and the other half to CLIME. Specifically, we use active learning to expand the training set with twenty-five target language documents and refine the embeddings by running CLIME on only twenty-five keywords. Then, we retrain a model using both the augmented training set and the refined embeddings.

4.3 Results and Analysis

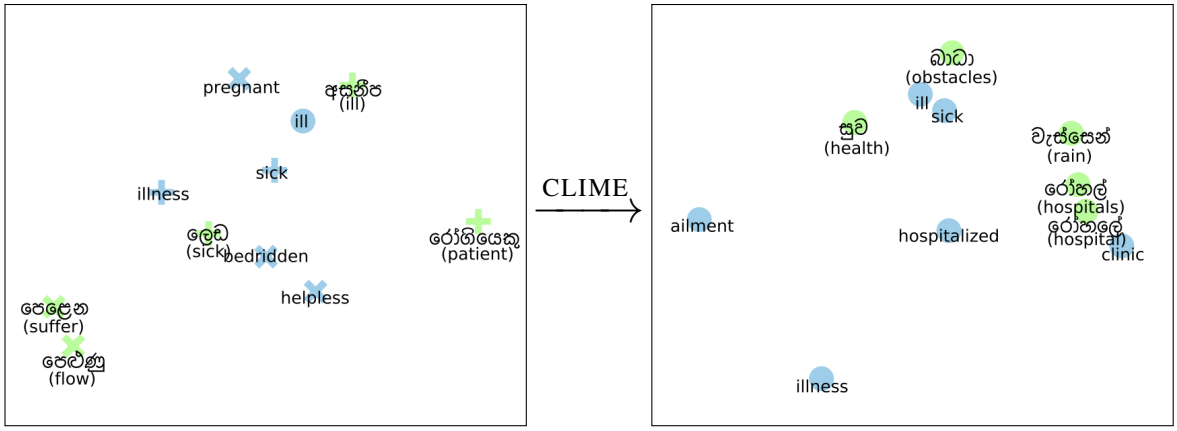
Effectiveness of CLIME. Figure 3 compares the four methods described in the previous section. CLIME is effective in this low-resource setting. On all four target languages, the classifier trained on embeddings refined by CLIME has higher accuracy than the classifier that trains on the original embeddings: CLIME reshapes embeddings in a way that helps classification. CLIME also has higher accuracy than active learning for most users. The combined method has the highest accuracy: active learning and CLIME are complementary. Single-sample t-tests confirm that **CLIME** is significantly better than **Base** and **A+C** is significantly better than **Active** (Appendix A.1).

Keyword Detection. We inspect the list of the fifty most salient keywords (Section 2.1). Most keywords have obvious connections to our classification task of detecting medical emergencies, such as "ambulance", "hospitals", and "disease". However, the list also contains some words that are unrelated to a medical emergency, including "over" and "given". These words may be biases or artifacts from training data (Feng et al., 2018).

[3] https://upwork.com/



(a) Neighborhood of "plague"



(b) Neighborhood of "ill"

Figure 5: T-SNE visualization of embeddings before (left) and after (right) CLIME updates. From one Sinhalese user study, we inspect two keywords, "ill" and "plague", and their five closest neighbors in English (blue) and Sinhalese (green). The Sinhalese words are labeled with English translations. Shape denotes the type of feedback: "+" for positive neighbors and "x" for negative neighbors.

Number of Keywords. To evaluate how feedback quantity changes accuracy, we vary the number of keywords and compare test accuracy on Tigrinya and Uyghur datasets (Figure 4). For each keyword s from one to fifty, we update the original embeddings using only the feedback on the top-s keywords and evaluate each set of embeddings with test accuracy. For both languages, test accuracy generally increases with more annotation at the beginning of the session. Interestingly, test accuracy plateaus and slightly drops after reaching an optimal number of keywords, which is around twenty for Tigrinya and about forty for Uyghur. One explanation is that the later keywords are less salient, which causes the feedback to become less relevant. These redundant constraints hamper optimization and slightly hurt test accuracy.

Qualitative Analysis. To understand how CLIME updates the embeddings, we visualize changes in the neighborhoods of keywords with t-SNE (Maaten and Hinton, 2008). All embeddings from before and after the user updates are projected into the same space for fair distance comparison. We inspect the user updates to the Sinhalese CCA embeddings (Figure 5). We confirm that positive neighbors are pulled closer and negative neighbors are pushed further away. The user marks "epidemic" and "outbreak" as similar to the keyword "plague", and these words are closer after updates (Figure 5a). For the keyword "ill", the user marks "helpless" as a negative neighbor, because "helpless" can signal other types of situations and is more ambiguous for detecting a medical emergency. After the update, "helpless" is pushed away and disappears from the nearest neighbors of "ill" (Figure 5b). However, a few positive neighbors have inadvertently moved away, such as the Sinhalese translation for "ill". The update algorithm tries to satisfy constraints for multiple

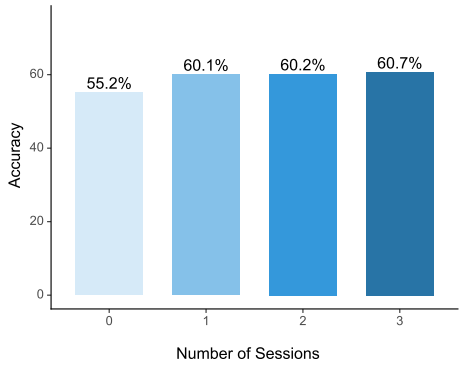


Figure 6: Progress of five Sinhalese users over three CLIME sessions. Largest increase in test accuracy occurs after first session. The leftmost point is the **Base** model from Figure 3. Average accuracy for first session is not the same as Figure 3 because only a subset of users are asked to complete three sessions.

keywords, so soft constraints may be overlooked. This motivates repeated CLIME sessions where the user can continue fixing errors.

4.4 Repeating User Sessions

We investigate the effects of having a user complete multiple CLIME sessions. After the user finishes a session, we fit the embeddings to their feedback, produce a new vocabulary ranking, and update the interface for the next session. We experiment on the Sinhalese dataset with CCA embeddings and ask five users to complete three sessions of fifty keywords. Average test accuracy increases with more sessions, but the improvement is marginal after the first session (Figure 6). By the end of the three sessions, one user reaches 65.2% accuracy, a significant improvement from the 55.2% baseline.

4.5 Comparing with Contextual Embeddings

Contextualized embeddings based on multilingual transformers reach state-of-the-art in many tasks, so we compare CLIME with these models. Most existing models (Wu and Dredze, 2019; Lample and Conneau, 2019) do not cover our low-resource languages. The only exception is XLM-R (Conneau et al., 2020), which covers Uyghur and Sinhalese. To compare with CLIME, we fine-tune XLM-R for three epochs with AdamW (Loshchilov and Hutter, 2019), batch size of sixteen, and learning rate of 2e-5. We compute average accuracy over ten runs with different random seeds.

For Uyghur, XLM-R has lower accuracy than our **A+C** approach (71.7% vs. 73.2%). This is impressive given that XLM-R uses much more resources: 270 million parameters, 2.5TB of multilingual Common Crawl data, and 500 GPUs. In contrast, the **A+C** model has 120K parameters and is built in less than two hours with a single GPU (including human interaction and model training).

For Sinhalese, XLM-R has higher accuracy than our **A+C** approach (69.3% vs. 63.7%). Common Crawl has much more Sinhalese words than Uyghur words. This aligns with our intuition: CLIME is more useful in low-resource settings, whereas multilingual transformers are more appropriate for languages with more data. Future work can extend the interactive component of CLIME to multilingual transformers.

5 Related Work

Cross-Lingual Word Embeddings. Ruder et al. (2019) summarize previous CLWE methods. These methods learn from *existing* resources such as dictionaries, parallel text, and monolingual corpora. Therefore, the availability and quality of training data primarily determines the success of these methods (Søgaard et al., 2018). To improve the suitability of CLWE methods in low-resource settings, recent work focuses on learning without cross-lingual supervision (Artetxe et al., 2018; Hoshen and Wolf, 2018) and normalizing monolingual embeddings before alignment (Zhang et al., 2019). In contrast, we design a human-in-the-loop system to efficiently improve CLWE. Moreover, previous CLWE methods are heavily tuned for the intrinsic evaluation task of dictionary induction, sometimes to the detriment of downstream tasks (Glavas et al., 2019; Zhang et al., 2020b). Our method is tailored for downstream tasks such as text classification.

Cross-Lingual Document Classification. Prior approaches transfer knowledge with cross-lingual resources, such as bilingual dictionaries (Wu et al., 2008; Shi et al., 2010), parallel text (Xu and Yang, 2017), labeled data from related languages (Zhang et al., 2020a), structural correspondences (Peter Prettenhofer, 2010), multilingual topic models (Ni et al., 2011; Andrade et al., 2015), machine translation (Wan, 2009; Zhou et al., 2016), and CLWE (Klementiev et al., 2012). Our method instead brings a bilingual speaker in the loop to *actively* provide cross-lingual knowledge, which is more reliable in low-resource settings. Concurrent to our work, Karamanolakis et al. (2020) also show that keyword translation is very useful for cross-lingual document classification.

Human-in-the-Loop Multilingual Systems. CLIME is inspired by human-in-the-loop systems that bridge language gaps. Brown and Grinter (2016) build an interactive translation platform to help refugee resettlement. Yuan et al. (2018) interactively align topic models across languages.

Active Learning. A common solution to data scarcity is active learning, the framework in which the learner iteratively queries an oracle (often a human) to receive annotations on unlabeled data. Settles (2009) summarizes popular active learning methods. Most active learning methods solicit labels for training examples/documents, while CLIME asks for word-level annotation. Previous active learning methods that use feature-level annotation (Raghavan et al., 2006; Zaidan et al., 2007; Druck et al., 2009; Settles, 2011) are not applicable to neural networks and CLWE. Closely related to our work, Yuan et al. (2020) propose an active learning strategy that selects examples based on language modeling pre-training.

Neural Network Interpretation. Our keyword detection algorithm expands upon prior work in interpreting neural networks. Li et al. (2016) uses the gradient of the objective function to linearly approximate salience of one dimension, which helps interpret and visualize word compositionality in neural networks. Their ideas are inspired by visual salience in computer vision (Simonyan et al., 2013; Zeiler and Fergus, 2014). We further extend the idea to compute the global salience of an entire word vector across a labeled dataset.

Specializing Word Embeddings. Our update equations modify prior work on specializing word embeddings that are designed to improve word embeddings with a large lexical knowledge base. Faruqui et al. (2015) *retrofit* word embeddings to synonym constraints. Mrkšić et al. (2016) expand the method by also fitting antonym relations. Mrkšić et al. (2017) includes both monolingual and cross-lingual constraints to improve CLWE. Glavaš and Vulić (2018) use a neural network to learn an specialization function that generalize to words with no lexical constraints. Closest to our work, Zhang et al. (2020b) retrofit CLWE to dictionaries and observe improvement in downstream tasks.

6 Conclusion and Future Work

CLIME is an interactive system that enhances CLWE for a task by asking a bilingual speaker for word-level similarity annotations. We test CLIME on cross-lingual information triage in international health emergencies for four low-resource languages. Bilingual users can quickly improve a model with the help of CLIME at a faster rate than an active learning baseline. Combining active learning with CLIME further improves the system.

CLIME has a modular design with three components: keyword ranking, user interface, and embedding refinement. The keyword ranking and the embedding refinement modules build upon existing methods for interpreting neural networks (Li et al., 2016) and fine-tuning word embeddings (Mrkšić et al., 2017). Therefore, future advances in these areas may also improve CLIME. Another line of future work is to investigate alternative user interfaces. For example, we could ask bilingual users to *rank* nearest neighbors (Sakaguchi and Van Durme, 2018) or provide scalar grades (Hill et al., 2015) instead of accepting/rejecting individual neighbors.

We also explore a simple combination of active learning and CLIME. Simultaneously applying both methods is better than using either alone. In the future, we plan to train a policy that dynamically combines the two interactions with reinforcement learning (Fang et al., 2017).

Acknowledgments

We appreciate the feedback from Joe Barrow, Shi Feng, Chen Zhao, and the anonymous reviewers. We thank Pedro Rodriguez, Jo Shoemaker, and Craig Harman for helping with the user study. Michelle Yuan is supported by JHU Human Language Technology Center of Excellence (HLTCOE). This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the BETTER Program contract #2019-19051600005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith.

2016. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.

Daniel Andrade, Kunihiko Sadamasa, Akihiro Tamura, and Masaaki Tsuchida. 2015. Cross-lingual text classification using topic-dependent word probabilities. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the Association for Computational Linguistics*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Deana Brown and Rebecca E. Grinter. 2016. Designing for transient use: A human-in-the-loop translation platform for refugees. In *International Conference on Human Factors in Computing Systems*.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

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

Gregory Druck, Burr Settles, and Andrew McCallum. 2009. Active learning by labeling features. In *Proceedings of Empirical Methods in Natural Language Processing*.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of Empirical Methods in Natural Language Processing*.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. *Conference of the North American Chapter of the Association for Computational Linguistics*.

Shi Feng, Eric Wallace, II Grissom, Mohit Iyyer, Pedro Rodriguez, Jordan Boyd-Graber, et al. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of Empirical Methods in Natural Language Processing*.

Goran Glavas, Robert Litschko, Sebastian Ruder, and Ivan Vulic. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the Association for Computational Linguistics*.

Goran Glavaš and Ivan Vulić. 2018. Explicit retrofitting of distributional word vectors. In *Proceedings of the Association for Computational Linguistics*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Yedid Hoshen and Lior Wolf. 2018. Non-adversarial unsupervised word translation. In *Proceedings of Empirical Methods in Natural Language Processing*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the Association for Computational Linguistics*.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of Empirical Methods in Natural Language Processing*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the European Chapter of the Association for Computational Linguistics*.

Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. 2020. Cross-lingual text classification with minimal resources by transferring a sparse teacher. In *Findings of EMNLP*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of Empirical Methods in Natural Language Processing*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of International Conference on Computational Linguistics*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proceedings of Advances in Neural Information Processing Systems*.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the Association for Computational Linguistics*.

Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. 2011. Cross lingual text classification by mining multilingual topics from wikipedia. In *Proceedings of ACM International Conference on Web Search and Data Mining*.

Benno Stein Peter Prettenhofer. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the Association for Computational Linguistics*.

Hema Raghavan, Omid Madani, and Rosie Jones. 2006. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7(Aug):1655–1686.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the Association for Computational Linguistics*.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Language Resources and Evaluation Conference*.

Burr Settles. 2009. Active Learning Literature Survey. Technical Report 1648, University of Wisconsin–Madison.

Burr Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of Empirical Methods in Natural Language Processing*.

Lei Shi, Rada Mihalcea, and Mingjun Tian. 2010. Cross language text classification by model translation and semi-supervised learning. In *Proceedings of Empirical Methods in Natural Language Processing*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the Association for Computational Linguistics*.

Stephanie Strassel and Jennifer Tracey. 2016. LORELEI language packs: Data, tools, and resources for technology development in low resource languages. In *Proceedings of the Language Resources and Evaluation Conference*.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Association for Computational Linguistics*.

Ke Wu, Xiaolin Wang, and Bao-Liang Lu. 2008. Cross language text categorization using a bilingual lexicon. In *International Joint Conference on Natural Language Processing*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of Empirical Methods in Natural Language Processing*.

Ruochen Xu and Yiming Yang. 2017. Cross-lingual distillation for text classification. In *Proceedings of the Association for Computational Linguistics*.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of Empirical Methods in Natural Language Processing*.

Michelle Yuan, Benjamin Van Durme, and Jordan Boyd-Graber. 2018. Multilingual anchoring: Interactive topic modeling and alignment across languages. In *Proceedings of Advances in Neural Information Processing Systems*.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*.

Mozhi Zhang, Yoshinari Fujinuma, and Jordan Boyd-Graber. 2020a. Exploiting cross-lingual subword similarities in low-resource document classification. In *Association for the Advancement of Artificial Intelligence*.

Mozhi Zhang, Yoshinari Fujinuma, Michael J. Paul, and Jordan Boyd-Graber. 2020b. Why overfitting isn't always bad: Retrofitting cross-lingual word embeddings to dictionaries. In *Proceedings of the Association for Computational Linguistics*.

Mozhi Zhang, Keyulu Xu, Ken-ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber. 2019. Are girls neko or shōjo? cross-lingual alignment of non-isomorphic embeddings with iterative normalization. In *Proceedings of the Association for Computational Linguistics*.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Cross-lingual sentiment classification with bilingual document representation learning. In *Proceedings of the Association for Computational Linguistics*.

A Appendices

A.1 Statistical Significance

Comparison	Model	p	t	df
CLIME vs. **Base**	SI(CCA)	<0.01	7.64	24
	SI(RCSLS)	<0.01	3.62	24
	IL(RCSLS)	<0.01	5.16	9
CLIME vs. **Active**	SI(CCA)	0.07	2.00	24
	SI(RCSLS)	<0.01	-7.09	24
	IL(RCSLS)	<0.01	3.96	9
A+C vs. **Active**	SI(CCA)	<0.01	4.297	24
	SI(RCSLS)	<0.01	3.40	24
	IL(RCSLS)	<0.01	13.97	9

Table 2: Results of single-sample t-tests between CLIME and **Base**, CLIME and **Active**, and **A+C** and **Active**, showing the p-value, the t statistic, and the degree of freedoms df. CLIME is significantly better than **Base**, and **A+C** is significantly better than **Active** across different languages and embedding models. The only combination with results that are not significantly different is CLIME and **Active** for Sinhalese (CCA).

We run single-sample t-tests with .05 significance level to see whether adding word-level annotations with CLIME can significally improve classification accuracy. We compare CLIME against **Base**, CLIME against **Active**, and **A+C** against **Active**. We use the user study results from the Sinhalese models (both CCA and RCSLS) and the Ilocano model. Table 2 shows that CLIME is not significantly different from **Active** for the Sinhalese CCA embeddings but does significantly improve accuracy for the Ilocano model. Overall, CLIME is significantly different from **Base** and **A+C** is significiantly different from **Active** across the experiments for all models.