Comparing *State-of-the-Art models for Part of Speech Tagging

Chibundum Adebayo[1] and **Hao-En Hsu**[2]
University of Stuttgart
3731211[1] and 3737244[2]

Abstract

In this study, we evaluate the effectiveness of various models for Part-of-Speech (POS) tagging by comparing a probabilistic approach, the Hidden Markov Model (HMM), with advanced deep learning methods, Transformer models and a Bidirectional Long Short-Term Memory with Conditional Random Fields (BiLSTM-CRF) ensemble. The HMM, serving as the baseline, utilizes a probabilistic framework to predict tag sequences. In contrast, the BiLSTM-CRF model captures contextual information and dependencies between output labels, while the Transformer model leverages self-attention mechanisms for rich syntactic and semantic information. Our experiments, conducted on datasets in English, German, Mandarin, and Afrikaans, demonstrate that deep learning approaches significantly outperform the HMM baseline in terms of F1 scores.

1 Introduction

POS Tagging is an essential pre-processing task for many natural language processing goals and applications (Martinez, 2012) which involves taking a sequence of words and assigning each word, its appropriate part of speech like NOUN or VERB (Jurafsky and Martin, 2024). Historically, POS tagging has evolved from rule-based approaches, which relied on handcrafted grammatical rules, to more advanced statistical models, such as Hidden Markov Models (HMMs) (Baum and Petrie, 1966), CRFs (Lafferty et al., 2001), and Maximum Entropy Markov models (MEMMs) which combines HMM and additional features from the Maximum Entropy model (McCallum et al., 2000).

In recent years, the field has shifted towards deep learning approaches for sequence labelling, employing models such as variants of Recurrent Neural Networks (RNNs): Uni-directional and Bi-LSTM networks, and Transformers and its variant, BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018).

In this paper, we systematically compare two deep learning approaches (Transfromer and BiLSTM-CRF, an ensemble model) to a probabilistic approach (HMM). The objectives of this paper is to:

1. Solve the limitations of HMMs by taking an end-to-end approach.
2. Compare the performance of these approaches across low, mid and high resourced languages.
3. Compare the impact of Transfer learning versus language-dedicated POS tagger.

2 Related Works

(Bi-)LSTM - CRF: BiLSTM-CRF model combines the strengths of BiLSTMs in capturing contextual information from both past and future tokens with the CRF's ability to model the dependencies between output labels. This ensemble model approach has achieved state-of-the-art (SOTA) in POS tagging (Huang et al., 2015), (Yasunaga et al., 2018). (Ma and Hovy, 2016) introduced a CNN layer for extracting character-level representation (BiLSTM-CNN-CRF) and (Akbik et al., 2019) extend contextual embeddings by incorporating character-level language models.

BERT: BERT utilizes a self-attention mechanism to simultaneously consider both left and right contexts, capturing extensive syntactic and semantic information. Fine-tuning BERT for part-of-speech (POS) tagging allows it to effectively disambiguate word meanings by leveraging its pre-trained embeddings. Variants of BERT, such as CamemBERT (Martin et al., 2019) and mybert (Madry et al., 2017), have achieved state-of-the-art performance. Specifically, (Martin et al., 2019) reported a 99.21 accuracy in French POS tagging with CamemBERT, while (Madry et al., 2017) achieved a BLEX score of 77.21 on the morphosyntactic analysis dataset.

3 Baseline Approach

We selected the Hidden Markov Model (HMM) as baseline sequence labelling algorithm. An HMM is a probabilistic model that computes a probability distribution over a set of labels and selects the most probable label sequence for a given observation sequence (Jurafsky and Martin, 2024). HMM models the relationships between the observations(sequence of words) and the hidden states(POS tags).

An HMM has four(4) components: a set of hidden states (POS tags), the emission probability matrix, the tag transition matrix and the initial probability distribution over each state, which is estimated as the probability of starting from each tag.

3.1 HMM Components

Emission Matrix: The emission matrix represents a sequence of observation likelihoods $P(word \mid tag)$, which is the probability of the word being emitted from the state corresponding to the part-of-speech tag. The probability can be derived using Maximum Likelihood Estimation (MLE):

$$P(w_i \mid t_i) = \frac{C(t_i, w_i)}{C(t_i)} \tag{1}$$

Tag Transition Matrix: The tag transition matrix computes the probability of a tag occurring given the previous tag $P(t_i \mid t_i - 1)$. The probability is computed using Maximum Likelihood Estimation (MLE) by counting from the corpus:

$$P(t_i \mid t_i - 1) = \frac{C(t_i - 1, t_i)}{C(t_i)} \tag{2}$$

Decoding with Viterbi Algorithm: The decoding process in HMM finds the most probable hidden states for a given observation sequence. This decoding process is done with the Viterbi algorithm. Decoding begins with initializing the Viterbi probability matrix of size (Number of tags, Input sequence length).

Each cell $v_t(j)$ in the Viterbi matrix is the probability of the HMM being in state j by passing through the most probable state sequence at each time step. Recursively filling each cell by taking the maximum over the Viterbi probability of being in each state from the previous time step, the tag transition probability from the previous tag to the current tag and the emission probability for the token given the current tag.

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t) \tag{3}$$

The most probable path sequence is extracted by backtracking from the final state and taking the most probable state for the observation O_t at each time step.

4 Advanced Approaches

Our baseline model exhibits several limitations. Firstly, it is unable to effectively manage long-term dependencies because HMMs depending only on the previous observation and tag. Secondly, it neglects contextual information, relying solely on probability calculations. Lastly, it cannot process rare or Out-Of-Vocabulary (OOV) words without creating additional features (morphological features such as suffix and prefix patterns, capitalization). To address these issues, we experiment with two end-to-end SOTA methods(Transformers and BiLSTM-CRF), eliminating the need for feature engineering.

We systematically compare the performance of the HMM, BiLSTM-CRF and Transformer models for tagging four (4) languages, including high-resourced (German, English), mid-resourced(Chinese) and a low-resourced (Afrikaans) languages.

4.1 Transformers

Transformers address key challenges in HMM by utilizing the self-attention mechanism, which handles long-term dependencies by allowing each token to attend to all others in the sequence, capturing the global context. This mechanism, combined with multi-head attention, captures various syntactic and semantic relationships within the sequence, effectively incorporating context. For rare or unseen words, transformers leverage pre-trained language models like BERT, which utilize extensive corpora to create robust word embeddings. Moreover, as BERT is pretrained on multiple languages, we aim to explore leveraging BERT models pretrained on high-resource language datasets to enhance POS tagging performance in low-resource languages, demonstrating potential improvements through knowledge transfer.

4.2 Bidirectional LSTM - CRF

Bi-LSTM-CRF model is an end-to-end approach, combining the strengths of Bi-LSTM and CRF to produce SOTA performance for sequence labelling.

Bi-Directional LSTMs: Bi-LSTM can capture contextual information by taking forward and

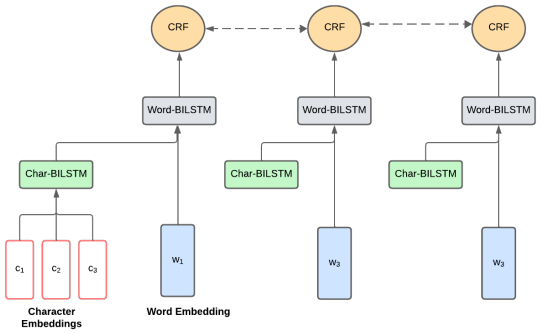

Figure 1: Architecture for Bi-directional LSTM + CRF model with the possibility of creating different model configurations

backward passes over the input sentence, using past and future features for the current time-step prediction. LSTMs maintain a memory using three multiplicative gates that determine the information to forget and pass to the next time-step, allowing for long-range dependencies and avoid the vanishing gradient problem in vanilla RNNs. Previous works extract morphological information by learning character-level representation of the input sequence. This additional information has proven to help with rare and Out-Of-Vocabulary words (Plank et al., 2016). The character representation for each word is feed along side its word embeddings to produce its final representation.

Conditional Random Fields: The Linear Chain CRF component provides the sentence-level dependencies between output labels. Similar to Bi-LSTM, CRF uses past and future tags to predict the current tag. Using only CRF for sequence labelling requires good knowledge of the task to develop hand-crafted features. The Bi-LSTM component learns the necessary features for the CRF layer.

CRF estimates a conditional probability distribution over the possible target sequence tags given the input sequence representation from the Bi-LSTM. Decoding is done with the Viterbi Algorithm, described in Section 3.1 to find the tag sequence with the most probable conditional probability (Ma and Hovy, 2016).

Our proposed method allows for different configurations of the model architecture by toggling the different components as in Figure 1.

	Train	Dev	Test	POS Tags
German	13,814	799	977	17
English	9,521	1,341	1285	17
Chinese	3,997	500	500	16
Afrikaans	1,315	194	425	16

Table 1: Corpus distribution showing: Number of unique sentences in each split and unique POS tags

5 Experimental Setup

5.1 Datasets

Baseline approach. We use the English POS data provided with the project task and it contains 48 unique tags.

Advanced approaches. We use the Universal Dependencies (UD) tree-bank datasets for the four(4) languages. The UD datasets are in CoNLL-U Format and each sentence line has the ID, word/punctuation, its lemma, Universal POS tag, language-specific tag, and other linguistic information.

To match the dataset structure from the baseline, we extract the token, lemma (if available) and its universal POS tag. The corpora distribution are shown in Table 1. No further pre-processing is performed on the datasets.

5.2 Implementation Details

5.2.1 Baseline Model

Our baseline project implements an HMM based model for sequence labelling. The model leverages probabilistic approaches, utilizing both emission and transition probabilities to determine the most likely sequence of POS tags, decoded by the Viterbi algorithm. The implementation is structured across several modules: `data module` extracts tokens and tags; `Evaluation Module` calculates the POS tagger's performance using precision, recall, and micro F1-score; `HMM Components modules` to construct and save the vocabulary, emission and tag transition matrices from the training data, and implements the Viterbi decoding algorithm; `Prediction-evaluation module` predicts POS tags for sentences and save predictions to a file for evaluation against gold standard tags; `Main module` orchestrates the execution of the HMM POS tagger and its performance evaluation.

For out-of-vocabulary words, we add the emission probability of the <UNK> token for the calculation in Equation 3. To avoid zero-probability outputs, we introduce a smoothing parameter to

shave off probability mass from non-zero probabilities.

5.2.2 Transformer

We implemented a BERT-based POS tagger for three languages: Mandarin, English, and German, utilizing pre-trained BERT models from Hugging Face. Specifically, 'bert-base-chinese' was employed for Mandarin, 'bert-base-uncased' for English, and 'bert-base-german-cased' for German. Training involved optimizing with the Adam optimizer using a fixed learning rate of 5e-5, while evaluation utilized micro F1 scores. Additional hyperparameters included a batch size of 8, number of epochs of 10 (and the best performance across these 10 epochs was recorded for analysis), and a dropout rate of 0.15. The models were saved and later used in transfer learning.

For our transfer learning experiments, we adapted pre-trained models from Mandarin, English, and German to Afrikaans, a lower-resourced language. We retained the original hyperparameters during this adaptation. We hypothesized that models transferred from English and German would perform better, as these languages, like Afrikaans, belong to the Germanic language family.

5.2.3 BiLSTM-CRF

We implemented the BiLSTM-CRF ensemble model using PyTorch and the PyTorch-CRF library (Kurniawan, 2019). Each example in the dataset is converted into its Tensor form on both word and character level using the `POS dataset` class. We also shuffle the reconstructed training data to avoid over-fitting or pattern memorization.

Pre-trained Word Embeddings. We use the FastText 300-dimensional pre-trained word embeddings for each language (Grave et al., 2018). The embeddings are downloaded and reduced dimension to 100 using the `FastTeXT Embed` module. Finally, we extract and save the embeddings for the vocabulary in the training data to a PyTorch tensor file.

Bi-directional LSTMs. The model architecture consists of two BiLSTM structures (Character-level and Word-level). The (Char-) Word LSTM is trained and then we apply a linear layer to map from the final output of the (Char-) Word LSTM to the tag set size. To predict the sequence labels without the CRF layer, a soft-max layer is applied for the probability distributions over the tags and returns the most probable tag sequence. The result is evaluated against the Gold standard tags using the Cross Entropy loss for weight optimization.

CRF. For the sentence-level labelling with CRF we pass the tag size for each language as input to the TorchCRF class, and the Bi-LSTM provides the contextualized features. We minimize the negative log likelihood loss of the tag sequence during training.

Parameter Optimization and Training. Different ensemble models can be trained with different configurations: use of Char-BiLSTM, pretrained word embeddings, and CRF layer by toggling the Boolean values in the `Model Parameters` module. We use embedding dimension of 25 for character and 100 for word. Every ensemble model is trained for 10 epochs with stochastic gradient descent (SGD) and momentum 0.9. A fixed learning rate of 0.015 did not perform well across the ensembles and languages and we experiment with different learning rates. The learning rate is updated after each epoch with a decay rate of 1e-6 using $\eta_t = \eta_0/(1 + decayrate)$ (Ma and Hovy, 2016).

Smoothing	0	0.0001	1.0
Dev	0.90	**0.929**	0.91
Test	0.91	**0.934**	0.91

Table 2: F1 score of HMM model with different smoothing values on Dev and Test sets

6 Results and Analysis

6.1 Baseline Model

The HMM with different smoothing values was evaluated using F1 scores on both development and test datasets, shown in Table 2 Without smoothing, the model achieved F1 scores of 0.90 and 0.91 on the development and test sets, respectively. Introducing a slight smoothing value of 0.0001 improved performance, yielding the highest F1 scores of 0.929 and 0.934 on the development and test sets. This improvement can be attributed to the mitigation of zero probabilities for unseen events, enhancing the model's robustness and generalization capability without overfitting. Conversely, increasing the smoothing value to 1.0 resulted in a slight decrease in F1 scores to 0.91 for both datasets, as excessive smoothing flattened the probability distributions too much, reducing the model's discriminative power.

6.2 Advanced Models

We evaluated the performance of two models, the BERT tagger and the HMM-based tagger, on the POS tagging task across English, German, Mandarin, and Afrikaans, with results reported as micro F1 scores in Table 4.

Transformer The BERT tagger demonstrated consistently high performance across both development and test datasets. It achieved micro F1 scores of 0.9176 (dev) and 0.9175 (test) for English, 0.9388 and 0.9200 for German, 0.7900 and 0.7957 for Mandarin, and 0.8825 and 0.8999 for Afrikaans. The BERT tagger outperformed the HMM tagger across all languages, with the most substantial improvements in German and English. This underscores BERT's robustness, effectiveness, and generalizability, particularly excelling in German while showing relatively lower performance in Mandarin.

However, for transfer learning, the results contradicted our hypothesis that the transferred model in German and English would perform better. Conversely, transfer learning in Mandarin performs the best among the three in both dev and test set, as table 5 shows. In order to look deeper into this phenomenon, we evaluated the transfer learning performance of pre-trained models from German, English, and Mandarin on Afrikaans, considering datasets of varying sizes (500, 1000, 3000, and 5000 samples). The results were reported in micro F1 scores as figure 2 shows. Initially, with 500 samples, the German model achieved the highest performance (0.6160), followed closely by Mandarin (0.6098) and English (0.6054). As the dataset size increased, all models showed improved performance. Notably, after 1000 training samples, the Mandarin model outperformed the others and achieved a score of 0.7662 with 5000 samples, despite our initial hypothesis favoring German and English due to linguistic similarity with Afrikaans. These findings suggest that while language family plays a role, the quality and characteristics of pre-trained models in different languages also significantly impact transfer learning effectiveness. Therefore, the results of transfer learning with different languages achieved similar F1 score.

BiLSTM-CRF. The different configurations of the proposed architecture in Figure 1 performed consistently across the four languages as shown in Figures 9 to 16. The configurations without character representation and CRF layers performed the worst compared to the other models but has better performance over the HMM baseline results in 4. Incorporating the character representations had significant improvement from ensembles without character-level information. The addition of the CRF layer had the second-best impact on the model performance. This observation shows that the CRF builds upon the good representation from the Bi-LSTM to further improve the performance. The impact of the pre-trained word embeddings varies across languages and showed the most improved with the low-resourced language (Afrikaans). The pre-trained word embeddings still remain a good starting point but learning task-specific embeddings directly has very comparable perfomance as seen in the results. Across all languages, the most complex configurations (Character-level with pre-trained embeddings for the word-level Bi-LSTMs and CRF Layer) has the best performance but with higher loss. The complex model requires more epochs to minimize the loss and achieve the S0TA performance. But due to limited GPU capacity and the number of models, we trained all models for 10 epochs.

	Learning Rate	Dropout	Dev	Test
German	1e-3	0.15	0.928	0.930
English	5e-4	0.30	0.916	0.906
Chinese	5e-3	0.15	0.885	0.886
Afrikaans	0.015	0.15	0.927	0.926

Table 3: Development and Test Micro F1 Score using the full architecture in Figure 1.

	English	German	Mandarin	Afrikaans
Dev	0.9176	0.9388	0.7900	0.8825
Test	0.9175	0.9200	0.7957	0.8999
HMM	0.772	0.806	0.75	0.852

Table 4: F1 score of BERT taggers in four different languages compared to baseline model HMM

	English	German	Mandarin
Dev	0.8396	0.8407	**0.8500**
Test	0.8585	0.8525	**0.8625**

Table 5: F1 score of Transfer Learning results on pre-trained models in three different languages

Figure 2: F1 scores on test sets for transfer learning on three languages with different number of tokens

7 Conclusion

In this paper, we addressed the limitations of Hidden Markov Models (HMMs) by implementing advanced end-to-end sequence labeling models, specifically focusing on deep learning architectures such as Transformer and Bi-LSTM-CRF. Our study involved a comparative analysis of these models across languages, ranging from low to high-resourced. Additionally, we examined the effectiveness of transfer learning compared to language-specific part-of-speech (POS) taggers.

Our findings demonstrate that the advanced end-to-end models significantly outperform traditional HMM-based approaches. By leveraging contextualized representations and incorporating morphological information, these models achieved superior performance on both seen and unseen words, without the need for data pre-processing, feature engineering, or task-specific knowledge. Notably, our language-dedicated models consistently outperformed the baseline, highlighting the advantages of shifting to deep-learning approaches in sequence labelling tasks.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59.

Leonard E Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 3483–3487.

Zhiheng Huang, Baidu Research, Wei Xu, and Kai Yu Baidu. 2015. Bidirectional lstm-crf models for sequence tagging.

Dan Jurafsky and James H. Martin. 2024. Speech and language processing (3rd ed. draft).

Kemal Kurniawan. 2019. pytorch-crf — pytorch-crf 0.7.2 documentation.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2:1064–1074.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Angel R Martinez. 2012. Part-of-speech tagging. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(1):107–113.

Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 591–598, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, pages 412–418.

Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:976–986.

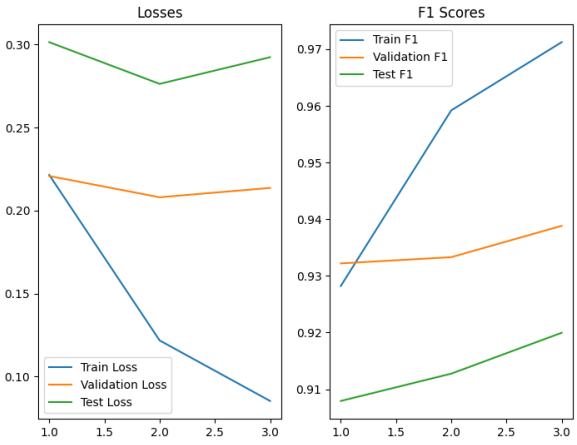

Figure 3: Loss and F1 score on BERT tagger for German data

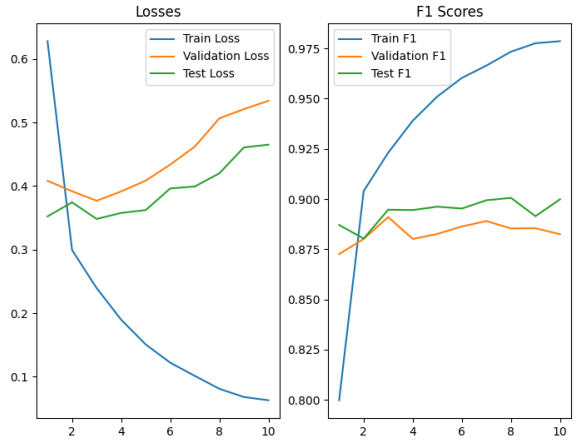

Figure 4: Loss and F1 score on BERT tagger for English data

A Appendix

Figures 3 to 8 are the results for BERT taggers. Figures 9 to 16 are the results for BiLSTM-CRF.

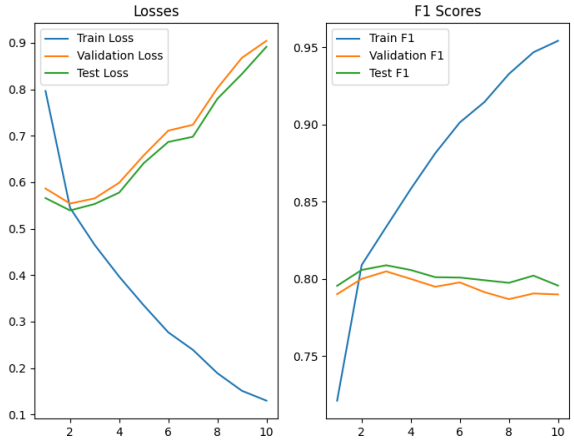

Figure 5: Loss and F1 score on BERT tagger for Mandarin data

Figure 6: F1 score on Transfer Learning in Afrikaans with pretrained German model

Figure 7: F1 score on Transfer Learning in Afrikaans with pretrained English model

Figure 8: F1 score on Transfer Learning in Afrikaans with pretrained Mandarin model

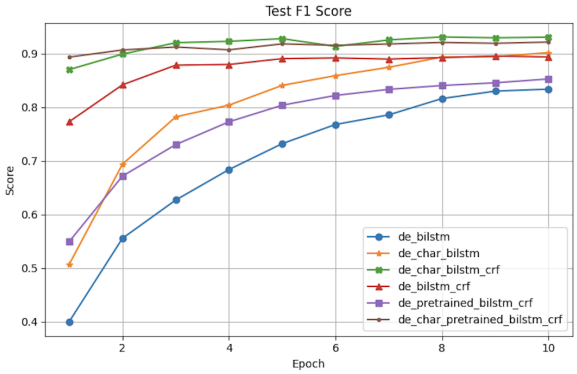

Figure 9: F1 score for German

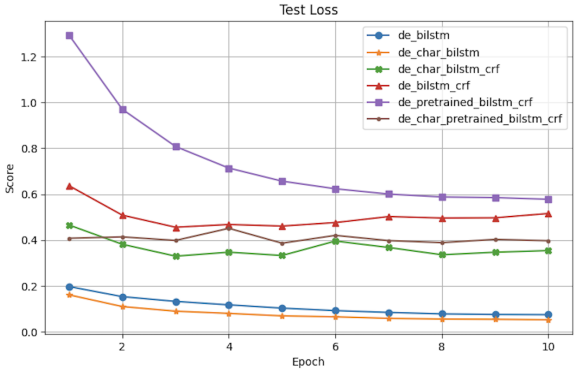

Figure 10: Test Loss for German

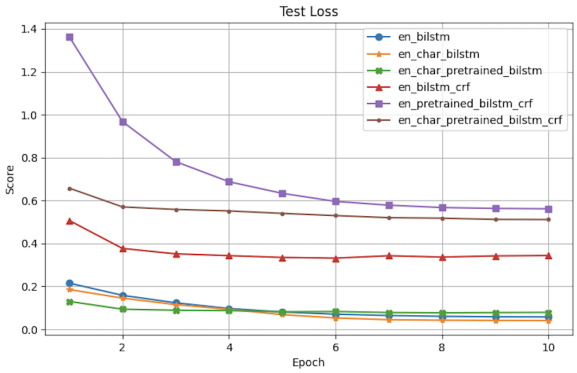

Figure 11: F1 score for English

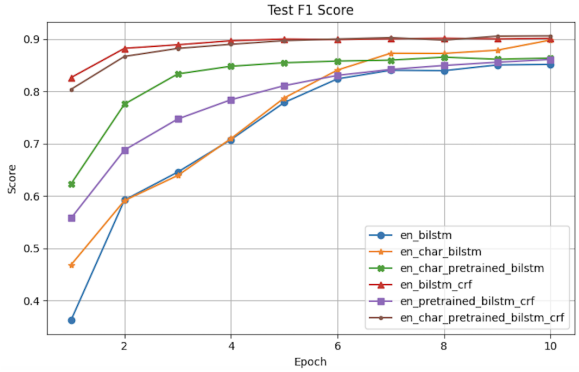

Figure 12: Test Loss for English

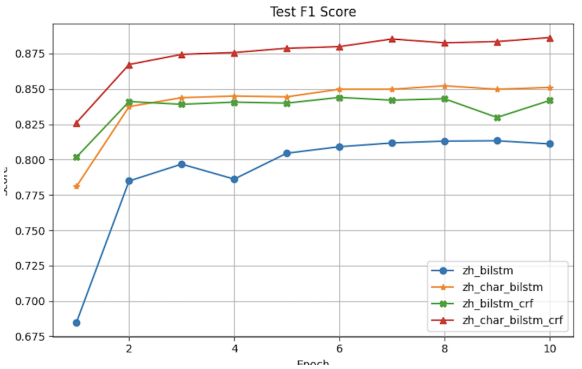

Figure 13: F1 score for Mandarin

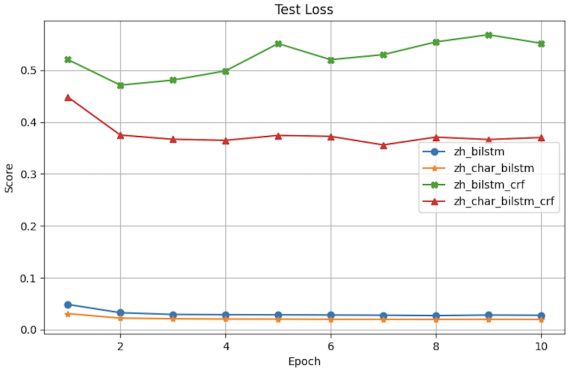

Figure 14: Test Loss for Mandarin

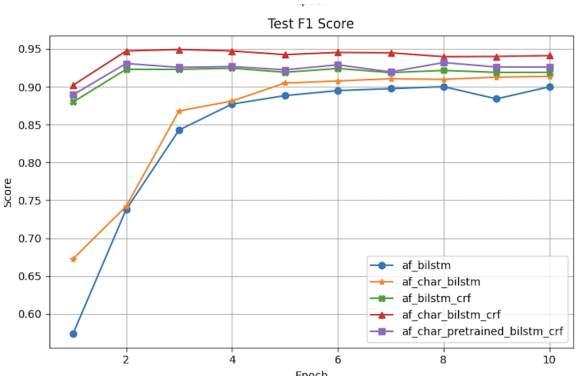

Figure 15: Test F1 Score for Afrikaans

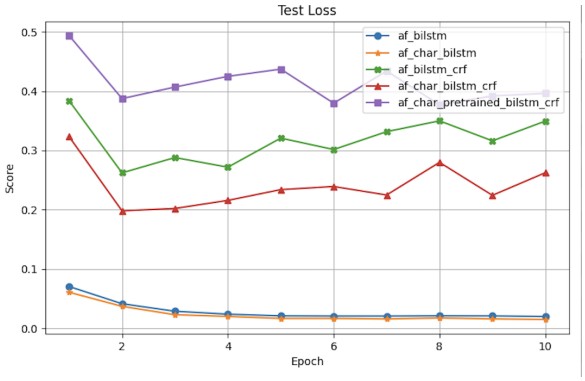

Figure 16: Test Loss for Afrikaans