Evaluating English Word Embedding Models for Part-of-Speech Tagging in Low-Resource Languages

Juhani Dickinson
jdickin9@uwo.ca

Paul Moore
pmoore44@uwo.ca

***Abstract*—Part-of-Speech (POS) tagging is a foundational task in Natural Language Processing (NLP) that provides essential syntactic information for applications such as text-to-speech, machine translation, and named entity recognition. This paper evaluates the effectiveness of various word embedding models like Flair, BERT, XLNet, T5, and a baseline static embedding model FastText, on POS tagging tasks. To isolate the impact of embedding quality, this study utilizes a standardized linear classifier trained on Universal Dependencies Treebanks data.**

This paper explores the cross-linguistic transferability of these models to low-resource languages by simulating a low-resource environment through the downsampling of a Norwegian dataset. The results indicate that transformer-based contextual embeddings significantly outperform the other models, with BERT, XLNet, and T5 all achieving F1 scores above 97% on English data. Additionally, it is found that relative model performance remains consistent across related languages (English to Norwegian), suggesting that high-quality English embeddings can be effectively transferred to related low-resource languages. T5 and XLNet emerged as particularly robust for cross-linguistic transfer, showing impressive performance on tagging sentences from the target language. These findings encourage the adoption of transformer-based embeddings to revitalize and support NLP development for endangered or low-resource languages.

***Index Terms*—pos tagging, part of speech, nlp, word embedding, contextual word embedding**

I. Introduction

A. *Parts of Speech*

In a sentence, each word has a syntactic role to play. In the sentence “The brown dog runs quickly towards food.”, “dog” denotes a thing, “brown” modifies or describes “dog”, “runs” denotes an action, and so on. These syntactic roles are commonly referred to as “parts of speech”, and are mapped to words using part-of-speech tags. However, this is not a one-to-one mapping. Many words have multiple meanings, and in many cases, these different meanings play different syntactic roles, mapping them to different parts of speech. For example, the word “orange”, spelled and pronounced identically, can be either a noun or an adjective, as seen in sentences (1) and (2).

“I often eat an orange after working out.” (Noun) (1)

“The orange car has arrived.” (Adjective) (2)

Figuring out what part of speech a word has is not an unsolvable task, however. Where a word is in the sentence and what words surround it play a major role in disambiguating between syntactic roles. In sentence (1), the word after “orange” is “after”, which is an adverb. “Orange” is either an adjective or a noun, and since the following word is an adverb, it cannot be an adjective. In sentence (2), the words surrounding “orange” are “the” and “car”, which are a determiner and a noun respectively. In this position, “orange” describes “car”, making it an adjective.

B. *Part-of-Speech (POS) Tagging*

The task of POS tagging (often simply called tagging) is as follows: given a sentence, label each word in the sentence with the POS tag that describes its syntactic role, seen below for sentence (3).

(3)	Can	you	spot	the	large	spot	on	your	nose	?
	AUX	D	VB	D	ADJ	N	PP	D	N	.

What makes the task of tagging challenging is the contextual requirement: without it, both instances of “spot” in sentence (3) would be tagged as either VB or N. Because of this, non-naive tagger implementations leverage some level of context to perform their task. Simple implementations like n-gram taggers assign tags to words based on both the word itself and the tags of the preceding n words.

C. *Word Embeddings*

A common issue with simple tagging implementations is resistance to parallelisation. Any operation below the level of sentence must be linear, as the tag of the current word depends on the tag(s) of the previous words. To get around the linear-time requirement, some taggers use word embeddings, which represent words as vectors. This embedding function takes into account the surrounding words, so that the context (but not necessarily meaning) of each word is stored in the embedding. Word embeddings are discrimitaive models usually trained using one of two methods: predict target word from context (Continuous Bag of Words, CBOW) and predict context from target word (Skipgram). In CBOW, the words surrounding the target word are mapped to one-hot vectors, which are each then processed by the classifier model and used to compute a single vector. This computed vector is then used to predict the target word. In Skipgram, the target word is mapped to a one-hot vector and then processed by the classifier model. This vector is then used to predict the context in which the target words appear, and is often done in parallel with other local words run through the same process. Using embeddings, tagging can be parallelised at the level of individual words.

1) Static Word Embeddings: While word embeddings take the surrounding words into account by design, static word embeddings take the naive approach to syntactic and semantic context. These mapping functions collect all contexts for a given word and uses them to generate a single embedding per word. Often, this approach fails to take into account that words can have different meanings ("river bank" vs. "bank account") or different syntactic roles. This means that a singular word embedding must hold information on every possible context for any given word, which causes problems when the different meanings of a word are not syntactically or semantically close to one another.

2) Contextual Word Embeddings: Contextual word embeddings solve the syntactic and semantic problems that static word embeddings have by moving from word embeddings to word token embeddings. Now, each instance of a word is a unique token that keeps all its context with it. Contextual embeddings map these tokens to vectors, preserving specific meaning. With contextual embeddings, a word like "spot" has as many tokens as there are unique occurrences of the word in text. What this ends up producing is clusters of vectors for the word "spot", with contexts like "Can you spot ..." mapping closer to each other and separate from a cluster of contexts like "... that large spot". While contextual embedding systems are more computationally demanding to train and use, their context-aware nature often leads to increased performance in context-sensitive tasks such as POS tagging.

II. Motivations

POS tags are useful and sometimes key pieces of information in many natural language processing (NLP) tasks such as text-to-speech (TTS), word-sense disambiguation, marking word order, and named entity recognition [1]. In TTS applications, the pronunciation of a word can change depending on what its syntactic role is or sometimes even what tense it is. The word "resume" is pronounced differently based on whether it is meant as "continue doing" or as "a document detailing professional experience", which can be easily disambiguated based on whether it acts as a verb or a noun. This disambiguation-by-tag is also used in word-sense disambiguation, like in sentence (3). In a search-engine setting, understanding whether "spot" refers to an activity or a visual marker allows for more accurate search results.

A. Benefits for NLP in Low-Resource Languages

While high-resource languages like English no longer use POS tagging in some popular NLP applications like AI assistants by making use of the massive collections of curated data now available, most languages do not have the luxury of large, high-quality datasets. In these low-resource languages, POS tags are still critical for NLP tasks [2]. For a task like machine translation, [3] found that using linguistic features derived from POS tags improved the translation performance not only between the low-resource language pair of Thai and Myanmar but also between Thai and English and Myanmar and English. While the highest accuracy was found on English-Thai translations, likely due to both languages having a Subject-Verb-Object sentence structure, the English-Myanmar and Thai-Myanmar pairs both saw improvements over the baseline.

While some low-resource languages have POS taggers or have the resources available for one to be reasonably made, many such languages do not. In such cases, POS taggers for high-resource languages are still quite useful. In low-resource languages where training data is scarce, one established method of finding training data is to use cross-linguistic transfer to automatically generate POS training data. This is done by lining up texts that appear in both a high-resource language and the target low-resource language and running POS taggers and parsers for the high-resource language [2]. The results from these taggers and parsers are then projected onto the low-resource text. With an advanced projection method such as Graph Label Propagation (GLP) [2] being used to transfer tags from only English, tagging accuracy on a variety of languages averaged 80.6%. Notably, methods such as GLP can leverage multiple parallel texts from high-resource languages to increase performance even further, with tests on the same variety of languages achieving an accuracy of 84.0%. However, leveraging multiple languages simultaneously is not always possible for certain low-resource languages due to the lack of available parallel data.

As well as being useful, the efficacy of cross-linguistic transfer is predictable. Reference [4] found that language families and morphological structures have a major impact on the performance of cross-linguistic performance. Specifically, performance was much higher when the source model for cross-linguistic transfer was in the same language family as the target language. This was especially true of morphologically-rich languages. Even outside of the same family, cross-linguistic transfer had better performance when moving between morphologically-rich languages, contrasting with morphologically-poor languages experiencing steeper drops in performance in similar cross-family settings. Importantly, no data from the target language was used for training or fine-tuning in these tests, which indicates that a target language being low-resource does not impact the efficacy of cross-linguistic transfer. This further indicates that embeddings trained on a high-resource source language contain information that is useful for a related target language, and therefore could achieve good POS tagging performance on the target dataset with minimal training data.

B. Benefits of an English POS tagger

Finding a good POS tagger for English, a high-resource language, has clear benefits for linguistically-related low-resource languages due to the success of cross-linguistic transfer. One example is Scots, a language in the Anglic family currently considered "vulnerable" by the UNESCO Atlas of the World's Languages in Danger [5]. Research suggests that NLP can support and help revitalise endangered languages [6] through machine translation, TTS, and integration into learning materials. With competent machine translation, more materials can be translated into Scots, which encourages learning and expands which facets of life can be interacted with in Scots. TTS has applications when paired with translation and learning tools, and brings one facet of accessibility into the language as well. Finally, machine-based language learning apps like Duolingo have made use of NLP and AI to provide more language learning content [7], and members of the Scots speaker community can make use of POS tagging directly to help with learning Scots linguistics.

To this end, we will test multiple prominent word embedding models head-to-head on the same English and Norwegian datasets and in order to determine the best model for POS tagging using cross-linguistic transfer.

III. Related Work

A. Existing Reviews of POS Taggers

Since POS tagging is a popular and foundational task to many NLP applications, many studies are done each year focusing on the challenges or potential new applications of POS tagging. References [8] and [9] conduct literature reviews of many POS tagging papers published between 2017 and 2023, looking at a variety of deep learning and machine learning approaches. The studies chosen for these reviews span a wide range of both high-resource and low-resource languages, and cover a wide range of topics, from sentiment analysis, subject-predicate recognition, and POS tagging in unusual environments such as social media posts.

A common issue with most of the studies is the lack of a standardised dataset for training or testing on. As a result, the findings from the studies reviewed cannot be directly compared, even between papers analysing the same language. Furthermore, there have not been any studies directly comparing the effectiveness of different word embedding models for use in POS tagging models. This serves as part of the motivation for our work here. Testing various word embeddings on a standard dataset enables direct comparisons of the embeddings used, as relative performance between NLP implementations is important for the development of the field.

Although many languages were represented in the studies selected for review, there was very little representation of the English language among the selection. Contemporary comparisons of POS taggers on English focus more on specific applications and more niche datasets. This serves as the other part of the motivation for our work here. Considering the importance of a good general POS tagger for work in low-resource languages, it is important to compare contemporary models on general use cases so new developments in the field of POS tagging are not left unconsidered for cases such as low-resource languages.

B. Studies on Cross-Linguistic Transfer

Discussed above in Section II, [4] provides crucial insights into the efficacy and patterns of cross-linguistic transfer. In this study, source-target language pairs were created from the chosen languages, and the XLM-RoBERTa multilingual transformer model was fine-tuned on each source language to serve as the POS tagger. Each pair was then analysed in terms of model performance, both quantitatively through precision, recall, and F1 scores, and qualitatively, through examining the areas where errors occurred to find commonly-appearing linguistic features such as morphology. However, this study only used BERT embeddings. While recreating this study using multiple embeddings is outside the scope of this work, we take inspiration from this by considering cross-linguistic transfer from English into Norwegian, both Germanic languages.

C. Word Embeddings Used

Based on the linguistic importance of syntactic and semantic context in POS tagging, our testing revolves around contextual word embeddings. To this end, we have selected 4 contextual word embeddings, with 1 static word embedding to serve as a measure of baseline performance. FastText was selected as the static embedding baseline and Flair, BERT, XLNet, and T5 were selected as the contextual embeddings due to their popularity, public availability, and performance.

1) FastText: FastText is a static word embedding developed by Facebook, designed to be fast to train on very large datasets and fast to use [10]. Its architecture is similar to CBOW, where it takes the word representations and averages them into a single text representation, which is then fed to a linear classifier, and softmax is used to compute the probability distribution. FastText was chosen for our work as our static embedding baseline for its ability to handle Out-of-Vocabulary (OOV) words, or words that were never seen in training. It does this by pooling embeddings for character n-grams instead of outputting word vectors directly. This is important for us as we perform tests on Norwegian as well as English, where nearly every word will be OOV.

Badri et al. [11] uses FastText in tandem with GloVe, another static word embedding, to detect offensive speech and hate speech. FastText was used for its ability to provide embeddings for OOV words, which can sometimes appear in hate speech and offensive speech.

2) Flair: Flair is the only non-transformer-based contextual word embedding we tested, as more recent popular and public contextual models are transformer-based [12]. The embedding portion of the Flair architecture consists of a character-based BiLSTM, which treats input text as a character sequence instead of a word sequence [13]. By using a bidirectional LSTM, the Flair embeddings captures syntactic and semantic context both from the start of the sentence to the first character in the word and from the end of the sentence to the last character. The final word embedding is the concatenation of the two character-based contexts.

Flair has been proven to be effective in applications that demand good out of vocabulary performance. For example, [14] creates a domain-specific Flair model called Clinical Flair, for use in Spanish-language clinical conditions. Due to "abundant misspellings" and medical words not seen in training found in clinical narratives, Flair was chosen due to its character-level processing, which, like FastText, is useful for handling OOV words.

3) BERT: BERT, which stands for Bidirectional Encoder Representations from Transformers, is a transformer-based contextual embedding model [15]. The architecture of BERT is a multi-layer bidirectional transformer encoder. In training, BERT is trained using a masked language model (MLM) task, where random tokens from the input sentence are obscured (masked) and the model is tasked with predicting the masked word based on the context. The bidirectional nature ensures that context is received from both sides of the masked word. Word embeddings can be read from BERT by extracting activations from the inner layers.

BERT has been used to advance the state-of-the-art in many NLP tasks such as [16], which uses BERT word embeddings for text representation in an automatic text summarization model.

4) XLNet: XLNet was developed in response to BERT, and claims that relying on input corruption via masking "neglects dependency between the masked positions" [17]. Another transformer-based contextual embedding model, XLNet's architecture uses two-stream self-attention, which uses both a query stream and a content stream during training. The query stream uses only the target position and not the target content to predict the token, and the content stream uses both the target position and content to predict the token. Reading activations from the content stream results in word embeddings.

5) T5: T5, which stands for Text-to-Text Transfer Transformer, is the most recently published contextual word embedding model used in this study. It takes a novel approach to text processing by treating every problem as a "text-to-text" problem [12]. T5 is an encoder-decoder transformer trained by self-attention. The encoder portion consists of a series of "blocks" consisting of a self-attention layer and a small feed-forward network, with scaling-only layer normalisation between each component inside each block. Word embeddings are obtained by using the outputs of the encoder.

IV. Methods

A. Hypotheses

We hypothesize that high-quality embeddings trained on a high-resource language can result in good POS tagging performance in a related low-resource language, and that high-quality embeddings for a high-resource language will be high-quality embeddings in low-resource languages. We also hypothesize that the powerful transformer based embeddings [12], [15], [17], [18] will result in superior performance on such tasks over more classical embedding methods [10], [13], [19]. Essentially, we will find the best embedding model for use in POS tagging of low-resource languages that are related to English.

B. Datasets

In order to meaningfully compare embeddings head to head, the classifiers using them to tag sentences must be trained and tested on the same datasets.

Each of the embedding models evaluated in this paper are trained on massive amounts of English data from various sources. Thus, we initially test them on the Universal Dependencies English Web Treebank dataset (UD_ENGLISH) [20], which contains sentences sourced from various forms of media. The Universal Dependencies English Web Treebank contains 16622 sentences that are split into three partitions: train, dev, and test, as illustrated in Table I.

In order to test the models' effectiveness at embedding text from low resource languages that are related to English, we use the Norwegian Dependency Treebank - Bokmål (UD_NORWEGIAN) [21]. However, Norwegian is not a low-resource language. Therefore we downsample the dataset to just 5% of it's original size, resulting in a corpus containing 1002 sentences partitioned as illustrated in Table I. This simulates a scenario where the data available for the target language is scarce, similar to how it is for real low-resource languages.

TABLE I
Distribution of Sentence Data

	English	**Norwegian**
Train	12544	785
Dev	2001	120
Test	2077	97

The Universal Dependencies Treebank datasets [22] are used for multiple reasons, including their diverse data sources and manageable size for local hardware. However, their most important features are their use of the Universal Part of Speech tagset (UPOS), designed to be a universal POS tag standard for multilingual applications, and that the annotations are hand corrected by humans to ensure 100% label accuracy.

C. Evaluation Metrics

To evaluate the taggers, we calculate macro precision, macro recall, and micro F1 score. Macro precision is the average precision of each of the N classes, weighted equally:

$$\text{Macro Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i}$$

Macro recall is the average recall of each class weighted equally:

$$\text{Macro Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i}$$

Macro precision and recall measure a model's performance on rare classes, uncommonly-seen syntactic roles such as interjections (UPOS tag `INTJ`). Micro F1 score is actually equivalent to the average accuracy in multi-class classification tasks:

$$\text{Micro F1} = \frac{2\sum_{i=1}^{N} TP_i}{2(\sum_{i=1}^{N} TP_i + \sum_{i=1}^{N} FP_i + \sum_{i=1}^{N} FN_i)}$$

Micro F1 score is used as a measure of overall performance.

D. Classifier Model Architecture

In order to isolate the embeddings in our tests, we use the same architecture to classify the embeddings from each embedding model. It is also critical that the performance of each model can be attributed to the embeddings rather than to the classifier itself, so a simple linear classifier is used. In each case, the linear classifier takes as input the embeddings and outputs a vector of size 17—one feature for each tag in the UPOS standard. Locked dropout with probability 0.5 is applied to the embeddings before classification.

E. Model Training

Following Schweter and Akbik [23], each model is trained using the Adam optimizer [24] and cross-entropy loss. The learning rate is tailored to each model, and adjusted automatically with a learning rate scheduler to ensure fast convergence without instability or over-fitting. For the transformer based embedding models, the final layer of the embedding model itself is fine-tuned alongside the linear classifier.

1) Framework: Akbik et al. [13] released a comprehensive NLP framework that leverages PyTorch [25] alongside their contextual string embeddings that are evaluated in this paper. The Flair framework is used to implement, train, and evaluate each of the embedding models. All training and evaluation was executed locally using CUDA on an NVIDIA RTX4060-mobile GPU.

2) Training on English Data: As a baseline for measuring the performance of each embedding model, we train over the English dataset. Each model is trained for 10 epochs with min-batch size of 16. Longer training stints could be considered, but given the relatively small quantity of data, too much could result in over-fitting. Moreover, as illustrated in Fig. 1, each model converges rather quickly (after only 4-6 epochs). This demonstrates that additional training would have diminishing returns. Figure 1 also indicates that the baseline static embedding model—FastText, struggles to converge. This is likely due to its inability to capture contextual information.

3) Training on Norwegian Data: Next, to compare performance in the high-resource language to the performance in the target language, each model is freshly trained on the downsampled Norwegian dataset, starting from the same initial state as the English models. Again, the models are trained for 10 epochs with min-batch size of 16, to avoid over-fitting. Figure 2 shows that each model converges nicely (apart from the static embedding model), as they do when trained on English data. Unsurprisingly, the development loss and accuracy do not reach the same levels as in Fig. 1.

4) Fine-Tuning English Models on Norwegian Data: Finally, while most of the heavy lifting of the models comes from the embeddings, we are interested to see if pre-training the simple classifier head on English data can give another performance boost. Therefore, we fine-tune the trained English taggers on the Norwegian dataset for an additional 5 epochs with a min-batch size of 16. The convergence of loss and accuracy can be seen in Figure 3.

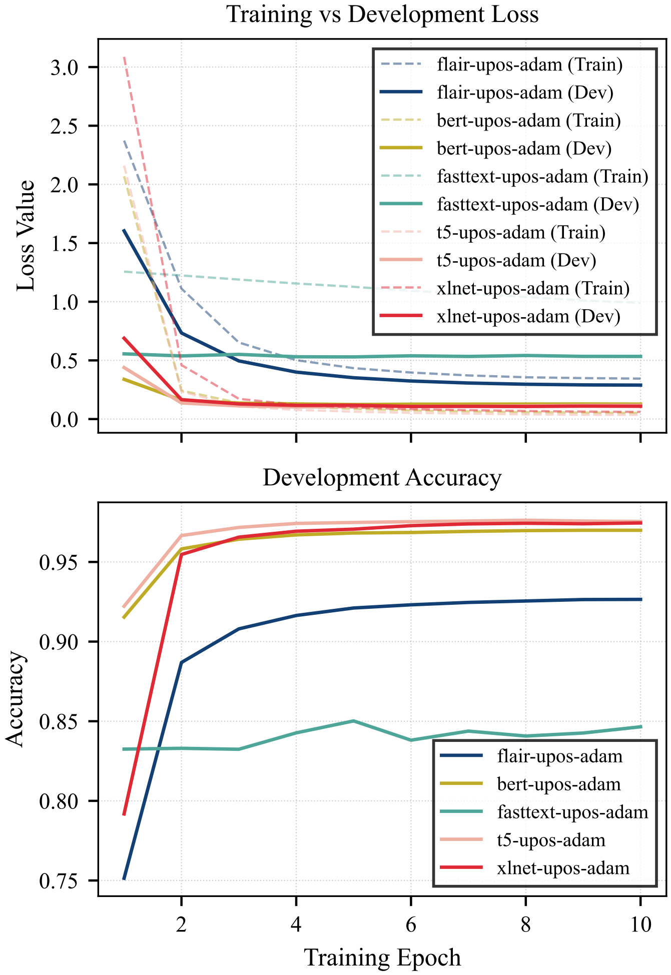

Fig. 1. Training vs. development loss and development accuracy per epoch on English data.

V. Results

A. English Model Performance

Rather unsurprisingly, the FastText embeddings yielded the worst performance, followed by the Flair embeddings, and then the three transformer based models performed similarly, each achieving an F1 score $>$ 97%. The macro precision, macro recall, and micro F1 score are shown for each model in Table II. The transformer models also perform quite well on the rare classes, as indicated by high macro precision and recall.

TABLE II
Performance of Taggers Trained on English

Embeddings	Precision	Recall	F1 Score
FastText	0.7944	0.7506	0.8471
Flair	0.8698	0.8079	0.9300
BERT	**0.9499**	0.9163	0.9714
XLNet	0.9486	**0.9177**	0.9745
T5	0.9116	0.9152	**0.9764**

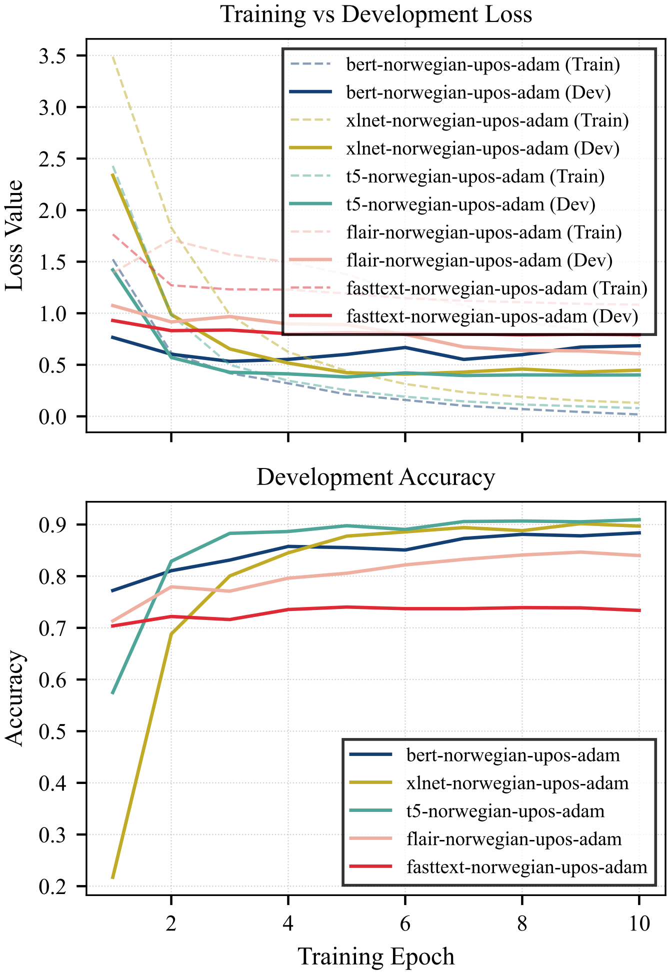

Fig. 2. Training vs. development loss and development accuracy per epoch on Norwegian data.

B. Norwegian Model Performance

Table III shows the results of training the taggers on Norwegian directly. Despite never being trained on Norwegian text apart from the limited training done in this study, both XLNet and T5 performed quite well on the Norwegian dataset. T5 was the most robust of the models, losing only approximately 0.06 on its F1 score. This indicates a strong ability to generalize to similar languages. All models also saw large drops in macro precision and recall, which can be attributed to the lack of rare class representation in a smaller dataset.

TABLE III
PERFORMANCE OF TAGGERS TRAINED ON NORWEGIAN

Embeddings	**Precision**	**Recall**	**F1 Score**
FastText	0.7104	0.6533	0.7162
Flair	0.8796	0.8678	0.8600
BERT	0.8408	0.8354	0.8894
XLNet	**0.8440**	**0.8507**	0.9070
T5	0.7565	0.7606	**0.9163**

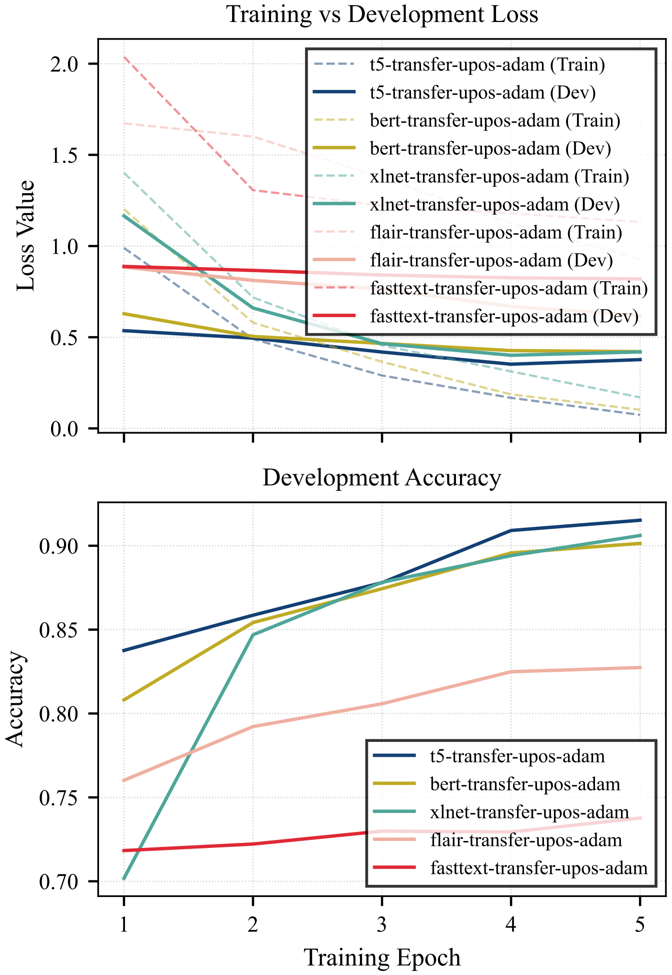

Fig. 3. Training vs. development loss and development accuracy per epoch during the fine-tuning of English models on Norwegian data.

C. Fine-Tuned English Model Performance

Finally, Table IV illustrates the results of fine-tuning the English taggers on the Norwegian dataset. Interestingly, despite its simplicity, pre-training the classifier head does appear to have a small performance benefit. Both T5 and XLNet see increases to their F1 score, 0.49% and 1.22% respectively. FastText also saw a large boost in performance of 3.59%.

TABLE IV
PERFORMANCE OF TAGGERS FINE-TUNED ON NORWEGIAN

Embeddings	**Precision**	**Recall**	**F1 Score**
FastText	0.8226	0.7021	0.7521
Flair	0.7755	0.7168	0.8264
BERT	0.8364	0.8481	0.8815
XLNet	0.7614	0.7652	0.9192
T5	**0.8710**	**0.8629**	**0.9212**

VI. Discussion

A. Embedding Quality

In our first experiment, we found that the transformer embeddings BERT, XLNet, and T5 were of the highest quality. When tested on an even playing field (The same classifier, training procedures, etc.), they significantly outperformed the non-transformer contextual embedding Flair, which itself significantly outperformed the static embedding baseline of FastText. This indicates that their contextual embeddings, especially transformer-based contextual embeddings, encode a deeper understanding of the underlying syntactic and semantic patterns of the English language, and are thus better suited to POS tagging tasks.

B. Generalizability

In our second experiment, we found that higher quality embeddings, and therefore higher POS tagging performance, on a high-resource language results in higher performance on a related low-resource language. The models using T5, XLNet, and BERT, who achieved very high accuracy on the English dataset, performed similarly relative to each other and the other models on the downsampled Norwegian dataset. This indicates that the transformer embeddings are capable of generalizing to languages that are related to English, including low-resource languages.

This easy generalizability is notable in itself, as it suggests that an embedding model that performs better than its competition in a high-resource language is likely to perform better than that same competition in a lower-resource language. This is especially useful for studies on other languages, as many studies that used word embeddings for work in other languages used static word embeddings, often Word2Vec [8], [9].

C. Pre-Training and Fine-Tuning

While not the focus of this paper, our third experiment revealed that pre-training the classifier head of the tagging model on the source language before fine-tuning the whole model on a target language can result in a performance increase even with the simplest possible classifier. It would not be crazy to assume that the benefits of such pre-training would be amplified by a more sophisticated classification model like many of the state-of-the-art taggers, who use a BiLSTM.

This performance increase for a related low-resource language indicates that linguistic similarities can be taken advantage of for cross-linguistic transfer even without access to a parallel corpus, which is very encouraging for work on low-resource languages that lack such corpora.

D. Which Embeddings Should Be Used?

It is clear that static embeddings should not be used for POS tagging in low-resource languages related to English, or really any POS tagger in general. Unless there are significant time or compute restrictions in place preventing the use of larger models like T5, static word embeddings are overshadowed. Even so, many large models like T5, XLNet, and BERT have been released with alternatives with fewer parameters for computationally-restricted environments.

For POS tagging in English, any of the transformer-based models can provide extremely rich embeddings capable of very high performance when partnered with a sufficient classifier. Even with the simplest possible classifier, the transformer embeddings reach near perfect performance (Table II). For POS tagging in other languages, transformer-based contextual embeddings should be prioritised over static embeddings where possible.

While T5, XLNet, and BERT demonstrated very similar performance when tested on English, BERT fell behind when evaluated on the simulated low-resource related language. Between T5 and XLNet, T5 was marginally more capable of generalizing to the new language.

E. Challenges and Limitations

The experiments resulted in some strange behaviour regarding the macro evaluation metrics. T5 performed quite poorly when trained on Norwegian, but then performed quite well when fine-tuned instead. Conversely, XLNet showed excellent macro precision and recall when trained directly on Norwegian, but its performance tanked when fine-tuned instead. See Tables III and IV. Since these metrics are measures of performance on rare classes, it is likely that this strange behaviour is a result of randomly donwsampling the Norwegian dataset to simulate a low-resource language. In some cases, the dataset contains sufficient rare class example for the model to learn, but other times there may not be any representation at all. This is a real and fundamental problem when working with low-resource languages and highlights a core limitation of using transfer learning for POS tagging in such languages.

With the rapidly evolving field of machine learning, new models and architectures are released each year. It is only a matter of time before the top performing models in this paper, like T5, will be outclassed in terms of embedding quality. T5 may currently be the best option for use in POS tagging of English, and its related low-resource languages, but this will not be the case forever. Research comparing today's models to the future state-of-the-art for use in this problem may be necessary as the field moves ever forward.

VII. Conclusion

In this paper, we study the POS tagging performance of popular word embeddings for English, as well as briefly examine the theoretical impacts of high-quality English word embeddings on POS tagging and NLP in other languages.

We found that that the transformer-based contextual word embeddings of BERT, XLNet, and T5 performed best with F1 scores all over 97%, suggesting that their embeddings contain the richest information for the task of POS tagging. Additionally, we found that contextual word embeddings in general greatly outperformed static word embeddings. This is consistent with linguistic theory, and suggests that properly accounting for and encoding context is important for many NLP tasks even beyond POS tagging.

Regarding the transfer of models between languages, we found that relative performance between embedding models stayed consistent when training embeddings and models on a related language. Furthermore, that this holds for a low-resource related language. This is promising for new work in these languages, and we encourage the use of transformer-based contextual embedding models on low-resource languages wherever possible. Furthermore, the consistent relative performance indicates that good English embeddings can be used as, or can be easily trained to be, good embeddings for related low-resource languages.

Somewhat consistent with existing literature cross-linguistic transfer of POS tags into low-resource languages, we found that training models and embeddings on English POS tags before fine-tuning the models for Norwegian POS tagging raised F1 scores for the best models, T5 and XLNet, when compared to their training results on Norwegian. Interestingly, BERT's performance on this task actually decreased compared to its Norwegian results. While the studies we found on cross-linguistic transfer used RoBERTa [2], [4], a variant of BERT compared to our use of base BERT, it is worth considering the use of T5 and XLNet in future studies on this topic.

A. Future Work

As mentioned in Section VI-E, the field of machine learning is constantly evolving. Studies like this one are always necessary to examine the new advances in embedding models, classifiers, and datsets, both in English (as tested here) and in other high-resource languages. In terms of short-term work, this study can be replicated using the same architectures on different languages, especially languages not related to English, which helps to generalise the relative performance findings to a wider variety of languages families. Considering the prevalence of LLMs in the past few years, it is possible that improvements in their models in the coming years will make them competitive in accuracy with the current state-of-the-art transformer embedding models.

Considering our baseline test on cross-linguistic transfer from English to a simulated low-resource Norwegian, there are many directions to go for future work on low-resource languages. In the short term, attempting to generalise these findings by replicating the methodology using a few low-resource languages as targets provides additional empirical evidence for the usefulness of a high-quality embedding such as T5 in low-resource contexts. In the longer term, studies such as [4] should be revisited either using the best-performing embedding to keep expected performance metrics up to date with the current state-of-the-art as well as to find any notable changes in performance between languages with certain (lack of) linguistic features. This can be further expanded to test multiple embeddings simultaneously, as what is a great embedding for cross-linguistic transfer in one language family may not be the best or even good at cross-linguistic transfer in other language families.

VIII. Acknowledgements

A. AI Use Acknowledgement

We did not use any LLMs to assist in writing this report.

B. Source code

We release all code used to train and evaluate the models publicly. These resources can be found at my github[1].

References

[1] S. Ghosh and B. K. Mishra, "Parts-of-speech tagging in nlp: Utility, types, and some popular pos taggers," in *Natural Language Processing in Artificial Intelligence*. Apple Academic Press, 2020, pp. 131–165.

[2] A. Imani, S. Severini, M. Jalili Sabet, F. Yvon, and H. Schütze, "Graph-based multilingual label propagation for low-resource part-of-speech tagging," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1577–1589. [Online]. Available: https://aclanthology.org/2022.emnlp-main.102/

[3] Z. Z. Hlaing, Y. K. Thu, T. Supnithi, and P. Netisopakul, "Improving neural machine translation with pos-tag features for low-resource language pairs," *Heliyon*, vol. 8, no. 8, 2022.

[4] A. Bankula and P. Bankula, "Cross-linguistic transfer in multilingual nlp: The role of language families and morphology," 2025. [Online]. Available: https://arxiv.org/abs/2505.13908

[5] C. Moseley, *Atlas of the World's Languages in Danger*. Unesco, 2010.

[6] S. Zhang, B. Frey, and M. Bansal, "How can NLP help revitalize endangered languages? a case study and roadmap for the Cherokee language," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1529–1541. [Online]. Available: https://aclanthology.org/2022.acl-long.108/

[7] "How duolingo uses ai to create lessons faster," https://blog.duolingo.com/large-language-model-duolingo-lessons/, accessed: 2025-12-18.

[8] A. Chiche and B. Yitagesu, "Part of speech tagging: a systematic review of deep learning and machine learning approaches," *Journal of Big Data*, vol. 9, no. 1, p. 10, 2022.

[9] N. Baruah and P. J. Goutom, "A comparative analysis of deep learning and machine learning for pos tagging," *Expert Systems with Applications*, vol. 288, p. 128026, 2025.

[10] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds. Valencia, Spain: Association for Computational Linguistics, Apr. 2017.

[11] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining fasttext and glove word embedding for offensive and hate speech text detection," *Procedia Computer Science*, vol. 207, pp. 769–778, 2022.

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[13] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proceedings of the 27th International Conference on Computational Linguistics*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1638–1649.

[14] M. Rojas, J. Dunstan, and F. Villena, "Clinical flair: A pre-trained language model for spanish clinical natural language processing," in *Proceedings of the 4th Clinical Natural Language Processing Workshop*, 2022, pp. 87–92.

[1] https://github.com/DNAPrototypeX/POS-Tagging-With-Word-Embeddings/

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[16] Q. Wang, P. Liu, Z. Zhu, H. Yin, Q. Zhang, and L. Zhang, "A text abstraction summary model based on bert word embedding and reinforcement learning," *Applied Sciences*, vol. 9, no. 21, p. 4701, 2019.

[17] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d 'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proceedings of Workshop at ICLR*, vol. 2013, 01 2013.

[20] N. Silveira, T. Dozat, M.-C. de Marneffe, S. Bowman, M. Connor, J. Bauer, and C. D. Manning, "A gold standard dependency corpus for English," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, 2014.

[21] L. Øvrelid and P. Hohle, "Universal dependencies for norwegian," in *International Conference on Language Resources and Evaluation*, 2016.

[22] M.-C. de Marneffe, C. D. Manning, J. Nivre, and D. Zeman, "Universal dependencies," *Computational Linguistics*, vol. 47, no. 2, pp. 255–308, 07 2021.

[23] S. Schweter and A. Akbik, "Flert: Document-level features for named entity recognition," *ArXiv*, vol. abs/2011.06993, 2020.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d 'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.