Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2736–2745
Marseille, 11–16 May 2020

Evaluating the Impact of Sub-word Information and Cross-lingual Word Embeddings on Mi'kmaq Language Modelling

Jeremie Boudreau,[1] Akankshya Patra,[2] Ashima Suvarna[1] and Paul Cook[1]
1. Faculty of Computer Science, University of New Brunswick
2. University of Southern California
jeremie@boudreau.me, patra@usc.edu, asuvarna31@gmail.com, paul.cook@unb.ca

Abstract

Mi'kmaq is an Indigenous language spoken primarily in Eastern Canada. It is polysynthetic and low-resource. In this paper we consider a range of n-gram and RNN language models for Mi'kmaq. We find that an RNN language model, initialized with pre-trained fastText embeddings, performs best, highlighting the importance of sub-word information for Mi'kmaq language modelling. We further consider approaches to language modelling that incorporate cross-lingual word embeddings, but do not see improvements with these models. Finally we consider language models that operate over segmentations produced by SentencePiece — which include sub-word units as tokens — as opposed to word-level models. We see improvements for this approach over word-level language models, again indicating that sub-word modelling is important for Mi'kmaq language modelling.

Keywords: Indigenous languages, language modelling, word embeddings

1. Introduction

Mi'kmaq is an Indigenous language spoken primarily in Eastern Canada (Johnson, 1996). It is polysynthetic and verb-oriented, and in the Eastern Algonquian language family. Mi'kmaq has roughly 8,000 speakers in Canada,[1] and is a low-resource language. There are Mi'kmaq dictionaries (Rand, 1888; DeBlois, 1996) and translated texts (DeBlois, 1990), but no large corpora. There has been very little prior computational work on Mi'kmaq, with the exception of Maheshwari et al. (2018), who built a web corpus of Mi'kmaq and carried out preliminary language modelling experiments using this corpus.

Language models are a crucial component for many language technology systems including spelling correction and word suggestion on smartphone soft keyboards. Mi'kmaq language modelling is, however, particularly challenging due to its rich morphology and the relatively small amount of data available. Mi'kmaq is polysynthetic, so each word is composed of many morphemes (Johnson, 1996). Rand (1888, p. iv) explains that a single Mi'kmaq word can essentially describe a whole English sentence. Language models are often trained on billions of tokens of text (Chelba et al., 2014; Merity et al., 2017; Jozefowicz et al., 2016), but the Mi'kmaq corpus built by Maheshwari et al. (2018) is only 76k tokens.

The rich morphology of Mi'kmaq suggests that language models that operate only at the word level, and do not model the internal structure of words, might perform poorly. Moreover, because of this rich morphology, we expect many out-of-vocabulary (OOV) words, and therefore it is important that a Mi'kmaq language model be able to handle OOVs.

In this paper we consider n-gram and RNN language models for Mi'kmaq. We tune these models over a range of parameters in an effort to establish a strong baseline. We then consider the use of pre-trained word embeddings to initialize the input layer of the RNN language models. We find that, even when trained on a small amount of data, fastText embeddings (Bojanowski et al., 2017) — which incorporate sub-word knowledge and are able to form representations for OOVs — give a substantial improvement.

Cross-lingual word embeddings (CLWEs) are methods to create word embeddings for multiple languages in the same vector space (Duong et al., 2016; Ruder et al., 2019). Adams et al. (2017) showed promising results incorporating cross-lingual embeddings into language models for some simulated low-resource languages. We further consider the use of CLWE methods to initialize our models, including the method used by Adams et al. (2017). We find that language models that incorporate cross-lingual embeddings do not perform better than models initialized with fastText embeddings.

The experiments described so far use a variation of perplexity for evaluation. We then consider an evaluation that measures the potential savings in terms of keystrokes when typing that is motivated by word suggestion on smartphone soft keyboards. In these experiments we again find that a language model incorporating fastText embeddings performs well. We further consider language models that operate over segmentations produced by SentencePiece,[2] which include subword units as tokens, in addition to the word-level language models considered so far. We see further improvements using SentencePiece, reinforcing the importance of subword modelling for Mi'kmaq language modelling.

2. Related Work

Our work draws on two major areas that are explored in this section. We first discuss language modelling techniques, and then discuss work on low resource languages, including applications of CLWEs.

[1] https://www12.statcan.gc.ca/census-recensement/2011/as-sa/98-314-x/98-314-x2011003_3-eng.cfm

[2] https://github.com/google/sentencepiece

2.1. Neural Network Language Modelling

Language models are a core component of many language technology systems, for applications such as spelling correction and next word prediction. Language modelling has traditionally been approached using n-gram models which work by counting sequences of n-grams in a corpus. Smoothing and back-off techniques can improve the performance of these models, and models using Kneser-Ney smoothing have shown strong results (Kneser and Ney, 1995; Chen and Goodman, 1999). More recently, neural network language modes have shown excellent results (Bengio et al., 2003; Mikolov et al., 2010). These models can incorporate contexts of arbitrary length, which can be much more powerful than simply counting fixed length word sequences (Goldberg, 2017). The first layer of these models is an embedding layer, which takes a high dimensional word vector and embeds it into a much smaller vector space (Goldberg, 2017).

Recurrent neural network (RNN) language models, in particular, have shown excellent performance because of their long memories (Mikolov et al., 2010). RNNs can be challenging to train because it is hard to back propagate gradients far into their memory, so RNN architectures with gated mechanisms to control the memory are often used (Goldberg, 2017). Long-short term memory (LSTM) models (Hochreiter and Schmidhuber, 1997) are a gated RNN architecture and have shown good results for English language modelling (Jozefowicz et al., 2015). Gated-recurrent unit (GRU) (Cho et al., 2014) networks are an alternate simpler architecture, and they have shown comparable results (Jozefowicz et al., 2015).

Weight initialization is an important consideration when training neural networks. Weights are often initialized using the uniform distribution (Glorot and Bengio, 2010; Mikolov et al., 2013b) or the normal distribution (He et al., 2015). Pre-training word embeddings, and using them to initialize a model's embedding layer, has been shown to improve language model performance (Bojanowski et al., 2017). These embeddings are typically trained using an algorithm such as continuous bag-of-words (CBOW) or skip-gram (Mikolov et al., 2013a; Mikolov et al., 2013b). FastText is a an implementation of these algorithms, and it has the advantage of also considering sub-word information to build embeddings (Bojanowski et al., 2017). FastText works by splitting words into character n-grams and learning vector representations for those n-grams. It forms the vector representation of a word by summing the representations of the character n-grams that compose it. These character n-grams can then be used to share sub-word representations between words, and we can build word representations for unseen words using their character n-grams.

Language models are often intrinsically evaluated using perplexity, which considers the probabilities given to sequences of words in a test corpus (Jurafsky et al., 2018). Better models obtain smaller values of perplexity. Perplexity (PPL) is calculated as follows (Jurafsky et al., 2018; Ueberla, 1994):

$$LTP = \sum_{i=1}^{N} log(P(w_i|w_1...w_{i-1})) \quad (1)$$

$$PPL = (2^{LTP})^{\frac{-1}{N}} \quad (2)$$

where N is the number of tokens.

Perplexity cannot be used to compare language models trained using different vocabulary sizes (Ueberla, 1994; Jurafsky et al., 2018). Adjusted perplexity is an alternate metric that applies a discount to out-of-vocabulary words, i.e., UNK tokens. This discount allows a fair comparison between models trained on different vocabulary sizes (Ueberla, 1994). Adjusted perplexity (APP) is calculated as follows:

$$ALTP = LTP - s \times log(r) \quad (3)$$

$$APP = (2^{ALTP})^{\frac{-1}{N}} \quad (4)$$

where s is the number of UNK occurrences and r is the number of UNK types (Ueberla, 1994).

2.2. Low Resource Languages

Language models for English and other high-resource languages are often trained on billions of tokens (Chelba et al., 2014; Jozefowicz et al., 2016); however, many languages have only much smaller corpora available, which makes language models harder to train. Transferring information from a high-resource language to a low-resource language is a technique that has proven useful in some NLP tasks (Duong et al., 2015; Adams et al., 2017). One of these techniques is the use of cross-lingual word embeddings (Ruder et al., 2019), which aims to build word embeddings for multiple languages in a common vector space. One method for building these embeddings is to use monolingual corpora from two languages and a bilingual lexicon (Duong et al., 2016). The approach suggested by (Duong et al., 2016) is an extension of the CBOW algorithm (Mikolov et al., 2013b), in which the target word in a context is replaced with its translation. A bilingual lexicon is used to do the translation. An approach based on the expectation maximization (EM) algorithm is used to pick the best translation when there are multiple translations for an entry in the lexicon.

Adams et al. (2017) relax the assumption of (Duong et al., 2016) that both target and source language corpora should be the same size, and use this technique for low-resource language modelling. They showed promising results on simulated low-resource languages, which were made by sampling high-resource languages. However, they failed to show positive results for Yongning Na, which is a real low-resource language. Adams et al. (2017) identify several issues related to the domain of their corpus (transcribed spoken narratives) and how the tones are structured in Yongning Na that could have contributed to their findings.

Maheshwari et al. (2018) carried out the first work in NLP focused specifically on Mi'kmaq. They constructed a Mi'kmaq web corpus of roughly 76k tokens. They then performed Mi'kmaq language modelling experiments in which

Corpus	Sentences	Tokens	Types
Training	5080	60k	18k
Dev	633	7.6k	1.6k
Test	633	8.1k	1.7k
Total	6346	76k	22k

Table 1: The number of sentences, tokens, and types in the Mi'kmaq corpora.

they considered n-gram models using KenLM (Heafield et al., 2013), a character-level RNN, and a word-level RNN that uses a CNN to incorporate character-level information (Kim et al., 2016). They found that the KenLM model performed the best out of the three. In this work we focus on Mi'kmaq language modelling, and further examine language models that incorporate sub-word information, and in addition consider language models that incorporate CLWEs.

3. Resources

In this section we describe the corpora and bilingual lexicon used in our experiments.

3.1. Corpora

We used the Mi'kmaq corpus built by Maheshwari et al. (2018). The corpus was already tokenized, and the only additional cleaning steps taken were normalizing various quote characters to ASCII single or double quotes, and swapping long hyphen characters with ASCII dashes. We randomly split the corpus into a training set of 5080 sentences, a dev set of 633 sentences, and a test set of 633 sentences. Table 1 shows the number of sentences, tokens, and types in the corpora. The training set is used for training language models. All preliminary experiments to select hyperparameters were performed on the dev data, and results are reported over the test data.

A corpus for a source language was needed to apply the CLWE approach suggested by Duong et al. (2016). We used English as the source language because it has the most entries in the bilingual lexicon we used (specifically Panlex, discussed in Section 3.2.). We used the latest Wikipedia dump as of 2018-12-02 as our English data.[3] We took a random sample of 5M sentences, and a random sample of 200k sentences, to use as source language corpora. Adams et al. (2017) used a source corpus of 5M sentences with a target corpus of 128k sentences. Our sample sizes were chosen so that we could consider the same source corpus size as they did, as well as the same ratio of source-to-target sentences.

3.2. Bilingual Lexicon

PanLex (Kamholz et al., 2014) was used by both Duong et al. (2016) and Adams et al. (2017) as the source of bilingual lexicons, so we used the same resource. PanLex is built by combining many translation resources, and includes entries for thousands of languages, including Mi'kmaq. Table 2 shows the number of single word translations from a source language into Mi'kmaq, for the top-5 languages with the most translations into Mi'kmaq in Panlex. We observe that English has many more entries than other languages, so we only performed experiments with English as the source language. Adams et al. (2017) showed that small lexicons negatively impacted performance, so it is important to try to get the largest possible lexicon.

Source Language	Number of Entries
English	4303
German	449
Italian	410
French	403
Japanese	377

Table 2: The number of single word translations from a source language into Mi'kmaq, for the top-5 languages with the most translations into Mi'kmaq in Panlex.

Out of the 4303 entries in the bilingual lexicon, 324 of the Mi'kmaq translations occur in the Mi'kmaq corpus. These 324 types correspond to 8301 tokens in the Mi'kmaq corpus.

4. Language Models

Since there has been little prior work done on Mi'kmaq language modelling, we first took steps to attempt to build a strong baseline model to compare other models against. Maheshwari et al. (2018) compared several approaches to Mi'kmaq language modelling, and showed that n-gram models performed the best of those considered, so we start by considering n-gram language models, and then proceed to consider word-level RNN models. We also consider different initialization schemes for the neural networks including the use of pre-trained monolingual fastText embeddings. Finally we consider models initialized with CLWEs.

4.1. N-gram Language Models

N-gram models were built with approximate Kneser-Ney smoothing using KenLM (Heafield et al., 2013). We considered n-gram orders from 2–6. We used the default settings for other parameters since they showed the best performance in preliminary experiments on the dev data.

4.2. RNNs

We considered two RNN model architectures: LSTM networks (Hochreiter and Schmidhuber, 1997) and GRU networks (Cho et al., 2014). We used PyTorch (Paszke et al., 2017), and its word-level RNN language model example as a base for implementing these models.[4] We tuned hyperparameters for these models including the number of layers, the amount of dropout, and the embedding size. We set the size of the hidden layer(s) to be the same size as the embeddings. We further considered the use of weight tying for the input and output layers (Inan et al., 2017; Press and Wolf, 2017), which has been shown to make language models much easier to learn.

[3] https://dumps.wikimedia.org/

[4] https://github.com/pytorch/examples/tree/master/word_language_model

Since using pre-trained monolingual embeddings, and CLWEs, is essentially a sophisticated way to initialize our language model, we additionally explored different initialization methods. By default, the weights of the layers were initialized using the uniform distribution with a range of $[-0.1, 0.1]$. We considered using the normal distribution with a mean of 0 and standard deviation of $\sqrt{\frac{2}{emb}}$, where emb is the size of word embeddings used (He et al., 2015). We additionally tested the uniform distribution scheme used by Mikolov et al. (2013a), and the Xavier initialization scheme proposed by Glorot and Bengio (2010). The ranges of these distributions are defined as $[-\frac{1}{2emb}, \frac{1}{2emb}]$ and $[-\frac{\sqrt{6}}{\sqrt{emb}}, \frac{\sqrt{6}}{\sqrt{emb}}]$ respectively. In all experiments we use the same initialization method for the input/output layers and hidden layers, unless otherwise noted.[5]

4.3. Monolingual Word Embeddings

FastText takes into account sub-word information in learning embeddings, and is able to form embeddings for out-of-vocabulary words. This could be well-suited for learning embeddings for a polysynthetic language such as Mi'kmaq where words have complex structure and many out-of-vocabulary words are expected to be encountered due to the rich morphology.

Mi'kmaq word embeddings were trained on the training data using both the skip-gram and CBOW models with 300 dimensions. We used a character n-gram size of 3–6 characters, and we used a minimum frequency of 5 occurrences. Both these parameters are default and showed the best performance in preliminary experiments. Training the embeddings on the training data resulted in embeddings for 1198 words. This fastText embedding model was used to initialize the first layer of our RNN models. For these models, the hidden layers were initialized using the normal distribution.

4.4. Cross-Lingual Word Embeddings

We considered two approaches to forming CLWEs, a direct approach, and the method of Duong et al. (2016).

For the direct approach, we use English word embeddings to initialize the first layer of the RNN. For every word in our Mi'kmaq vocabulary that has a match in the bilingual lexicon, we use the corresponding English word embedding as the embedding for the Mi'kmaq word. We use 300 dimensional fastText embeddings pre-trained over English Wikipedia as our English word embeddings.[6] We refer to this method as Direct CLWE.

In order to explore the impact of having the correct English embedding, as opposed to just having an English embedding, on the Direct CLWE approach, we considered several other approaches. The first method, referred to as Direct RAND_TRANS, randomly selects an English embedding for every Mi'kmaq word that has a translation in the bilingual lexicon. The second method, Direct RAND, randomly selects an English embedding for every Mi'kmaq word in the vocabulary.

Duong et al. (2016) provide an implementation to build CLWEs.[7] We built embeddings using this approach with both English corpora (5M sentences and 200k sentences) as the source language, using the default window size of 48, which Adams et al. (2017) argue mitigates word reordering effects. We refer to these approaches as Duong 5M and Duong 200k.

For all methods described in this subsection, if the embedding for a Mi'kmaq word is not initialized using the cross-lingual approach, the weights for those words are initialized using the normal distribution. For Direct CLWE and Direct RAND_TRANS, this occurs in the case of words that are not in the bilingual dictionary. For the Duong 5M and 200k approaches, this occurs for words that are out-of-vocabulary with respect to the learned cross-lingual embeddings. The hidden layers are also initialized using the normal distribution.

5. Results

In this section we first present results using n-gram and RNN language models, considering a range of parameter settings, to establish a strong baseline. We then consider initializing the input layer of the RNN language models with pre-trained monolingual embeddings. Finally we consider approaches that incorporate cross-lingual word embeddings.

5.1. Baseline Language Models

In Table 3, we compare KenLM and RNN language models, for a range of parameter settings.[8] We treat these models as baselines for subsequent experiments.

We observe that the best KenLM model ($n = 4$) has an adjusted perplexity of 2410.53. This performance is much worse than even the worst RNN model. This difference appears to be due to the large amount of probability mass assigned to UNK by KenLM.

The best GRU model, and the best LSTM model, on both dev and test, use 1 hidden layer, a dropout rate of 0.5, and an embedding layer of size 300. We therefore choose these two models for further experiments. The GRU model with these parameter settings achieves the lowest adjusted perplexity overall on both datasets.

For the results in Table 3, weights were initialized using the uniform distribution with a range of $[-0.1, 0.1]$. In Table 4 we consider alternative approaches to initializing these weights, specifically the approach used by Mikolov et al. (2013a), the Xavier scheme of Glorot and Bengio (2010), and the normal distribution (He et al., 2015) (discussed in Section 4.2.). We see a noticeable improvement with the use of the normal distribution to initialize the weights for

[5] In preliminary experiments we considered using different initialization methods for the input/output layers and hidden layers, but found this did not lead to improvements.

[6] `https://fasttext.cc/docs/en/english-vectors.html`

[7] `https://github.com/longdt219/XlingualEmb`

[8] For the RNN models, we first tuned the number of hidden layers, then the number of dimensions in the input and output layer, and then the dropout rate. We did not explore all combinations of parameter settings. In preliminary experiments we considered the use of weight tying for the RNN language models, and found this to give lower perplexity. We therefore only report results for RNN language models that use weight tying.

Model	Hyperparameters	Adjusted Perplexity	
		Dev	Test
KenLM	$n = 2$	2501.18	3094.59
	$n = 3$	1962.85	2489.9
	$n = 4$	**1900.67**	**2410.53**
	$n = 5$	1902.43	2413.81
	$n = 6$	1910.58	2421.89
GRU	$l = 2, r = 0.2, d = 200$	933.63	1031.85
	$l = 1, r = 0.2, d = 200$	810.32	896.44
	$l = 1, r = 0.2, d = 100$	830.44	913.51
	$l = 1, r = 0.2, d = 300$	771.08	862.81
	$l = 1, r = 0.5, d = 300$	**700.42**	**768.68**
LSTM	$l = 2, r = 0.2, d = 200$	974.41	1100.65
	$l = 1, r = 0.2, d = 200$	833.56	913.91
	$l = 1, r = 0.2, d = 100$	861.09	960.86
	$l = 1, r = 0.2, d = 300$	816.33	890.79
	$l = 1, r = 0.5, d = 300$	**755.70**	**827.68**

Table 3: Adjusted perplexity for baseline language models. n represents the n-gram order for n-gram models, l represents the number of hidden layers in the model, r represents the dropout rate used, and d represents the number of dimensions in the input and output layer. The best adjusted perplexity, for each model type, on each dataset, is shown in boldface.

Model	Initialization	Adjusted Perplexity	
		Dev	Test
GRU	Uniform	700.42	768.68
	Mikolov	857.61	954.50
	Xavier	778.49	870.28
	Normal	**619.98**	**674.67**
LSTM	Uniform	755.70	827.68
	Mikolov	884.55	974.50
	Xavier	812.75	878.60
	Normal	**672.09**	**726.49**

Table 4: Performance of the RNN models with different initialization methods. The best adjusted perplexity, for each model type, on each dataset, is shown in boldface.

Model	Initialization	Adjusted Perplexity	
		Dev	Test
GRU	CBOW	548.09	602.57
	Skip-gram	**486.49**	**535.73**
	Word-level skip-gram	578.98	638.67
LSTM	CBOW	569.65	623.84
	Skip-gram	**483.13**	**537.00**
	Word-level skip-gram	560.07	616.07

Table 5: Performance of RNN models with monolingual FastText embeddings used to initialize the embedding layer. The best adjusted perplexity, for each model type, on each dataset, is shown in boldface.

both the GRU and LSTM models. We consider both the GRU and LSTM models with weights initialized using the normal distribution as baselines for further experiments.

5.2. Monolingual Word Embeddings

In Table 5 we consider the use of pre-trained monolingual word embeddings to initialize the embedding layer of Mi'kmaq language models. Using fastText embeddings trained with either CBOW or skip-gram gives substantial improvements over the baseline approaches. For both the GRU and LSTM, on each dataset, skip-gram gives the best performance. Interestingly, although the GRU models generally performed better than the LSTM models when using random initialization schemes (Table 4), here the performance of the GRU and LSTM initialized with skip-gram embeddings is quite similar.

As a point of comparison we also consider word-level skip-gram embeddings that do not incorporate sub-word information ("Word-level skip-gram" in Table 5). These models perform worse than the corresponding skip-gram models that incorporate sub-word information. These findings suggest that sub-word information is important for Mi'kmaq language modelling.

5.3. Cross-Lingual Word Embeddings

Table 6 compares the CLWE approaches with previous models using the uniform and normal distributions for initialization, and initialization using skip-gram embeddings.[9] We first note that all of the direct approaches outperform those using the Duong methods. Moreover, the Duong methods performed much worse than the models using uniform initialization. The experiments performed by Adams et al. (2017) showed that this CLWE approach was not useful for simulated low-resource language modelling for a target corpus smaller than 32k sentences. We only have 5k training sentences, so there are likely not enough examples for the model to learn good embeddings using this technique. Adams et al. (2017) simulated different lexicon sizes, and showed that a lexicon size of about 10k entries was the critical point for achieving good performance. Our

[9] All experiments in this paper used the same random seed to initialize the models. To consider the impact of random seeding, we re-trained each model in this sub-section 5 times with different random seeds, and calculated the average perplexity across the 5 runs. The findings were overall consistent with those reported in this sub-section.

Model	Initialization	Adjusted Perplexity	
		Dev	Test
GRU	Direct CLWE	621.44	673.01
	Direct RAND_TRANS	626.14	681.28
	Direct RAND	682.01	743.83
	Duong 5M	1148.54	1244.43
	Duong 200k	876.30	980.25
	Uniform	700.42	768.68
	Normal	619.98	674.67
	Skip-gram	**486.49**	**535.73**
LSTM	Direct CLWE	689.64	743.37
	Direct RAND_TRANS	659.23	713.81
	Direct RAND	660.33	701.62
	Duong 5M	819.59	933.03
	Duong 200k	786.21	889.21
	Uniform	755.70	827.68
	Normal	672.09	726.49
	Skip-gram	**483.13**	**537.00**

Table 6: Adjusted perplexity of RNN models with various CLWE methods, as well as select results from previous subsections for comparison. The best adjusted perplexity, for each model type, on each dataset, is shown in boldface.

lexicon has only 4303 entries, and so the poor performance in this case is likely due to the small lexicon size. Furthermore, only 324 of the Mi’kmaq words in these entries occur in our corpus, and only 126 of these words have a frequency of 5 or greater, which is the cutoff for learning cross-lingual embeddings in this approach.[10] These findings indicate that there is a lack of cross-lingual examples to produce useful embeddings.

Turning to the direct approaches, we see that the best CLWE approach is the GRU with Direct CLWE, which achieves an adjusted perplexity on the test data of 673.01. In the case of Direct RAND_TRANS, the results are worse than Direct CLWE for the GRU, but not for the LSTM. This inconsistency suggests that the Direct CLWE approach is unable to effectively make use of information transferred from the source language. Moreover, these approaches are, at best, roughly on par with using the normal distribution for initialization. This is perhaps unsurprising because, for these CLWE approaches, words that don’t get an embedding via translation are initialized with the normal distribution, and the bilingual lexicon is relatively small, as noted above.

Direct RAND performs better than initialization using the uniform distribution for the GRU, and better than initialization using either the uniform or normal distribution for the LSTM. This result seems to indicate that a direct transfer approach might be useful because of the vector structure of the English embeddings, as opposed to the information that was transferred from the source language via the bilingual dictionary.

The best models in our experiments are the GRU and LSTM models initialized with fastText skip-gram embeddings that use sub-word information. The performance difference compared to simpler models was large (the skip-gram initialized GRU had an adjusted perplexity of 535.73 on the test data, while the GRU initialized with the normal distribution had an adjusted perplexity of 674.67). We suggest two main reasons for this difference. First, Mi’kmaq is a polysynthetic language, and this means that a model that considers sub-word information, as an approximation to morphology, might be critical. Second, this method can be used to build embeddings for out-of-vocabulary words, which is important because there is a large number of out-of-vocabulary words in the test data.

6. Edit Distance and Mi’kmaq Orthographies

In an attempt to improve the Direct CLWE method we explored a few ideas to try to increase the number of matches between words in the corpus and entries in the lexicon by making this matching less strict.

We first built a model where the matching was based on Levenshtein distance (Jurafsky et al., 2018). If we consider the Direct CLWE approach in terms of Levenshtein distance, the distance between a word in the corpus, and a word in the lexicon, must be 0 for them to match. However, there are variations in Mi’kmaq orthographies (Battiste, 1985), and the corpus is potentially noisy because it is a web corpus. Allowing for less strict matching could therefore potentially lead to improvements. Allowing for matches with a Levenshtein distance of 0 or 1, 1268 types in our corpus match a lexicon entry, as opposed to 324 when using exact match (i.e., a Levenshtein distance of 0). However, the adjusted perplexity of this approach using a GRU was 676.29 on the test data, which was worse than the original Direct CLWE approach (673.01, result shown in Table 6).

We further considered an approach that incorporates knowledge of differences in Mi’kmaq orthographies. Two contemporary Mi’kmaq orthographies are Francis/Smith and Listuguj. One difference between these orthographies is their representation of a velar stop, where Francis/Smith uses *k* while Listuguj uses *g*.[11] In this approach we applied a normalization step to treat these characters as equivalent, and then applied the Direct CLWE approach (i.e., with a Levenshtein distance of 0 for matching words in the lexicon and corpus). This resulted in 675 types in our corpus matching a lexicon entry, and gave an adjusted perplexity of 674.95 on the test data, which is also worse than the original Direct CLWE method. Nevertheless, because a single Mi’kmaq orthography uses only one of *k* or *g*, such a normalizing step might still be important to consider in future work on Mi’kmaq

7. Keystroke Savings Evaluation

The experiments so far have been based on intrinsic evaluation using adjusted perplexity. In this section we consider an evaluation motivated by the task of next word suggestion, for example as is common with smartphone soft key-

[10] We performed additional experiments with a cut-off of 1, but this modification showed no improvement.

[11] These orthographies also vary with respect to their representation of vowel length.

boards. In the following subsections we describe the models considered and the evaluation methodology, and then present results.

7.1. Models

In these experiments we consider the GRU language model in two settings: 1.) with weights initialized using the normal distribution, and 2.) with the input layer initialized using monolingual fastText embeddings. The former we consider as a baseline, and the latter is the model found to perform best so far.

In addition, because of the finding that subword information appears to be important in Mi'kmaq language modelling, we consider language models that operate at the subword level. We use SentencePiece[12] to tokenize our corpora using a fixed size vocabulary. Under this tokenization scheme tokens can correspond to subword units. We apply the same GRU language model (i.e., with the same parameter settings) as for the models that are based on word-level tokenization. When using SentencePiece, we also consider the two approaches to initializing the weights that we consider for the word-level language models, i.e., initializing weights using the normal distribution, and using fastText embeddings to initialize the input layer. In the case of the fastText embeddings, we consider training fastText using both word-level tokenization (i.e., the same as for previous fastText models), and using tokenization based on SentencePiece. In both cases we use the same fastText settings as in our previous experiments.

7.2. Evaluation

Smartphone soft keyboards often provide suggestions for the next word or word being typed, that the user can select instead of typing the word in its entirety. In these experiments we evaluate language models using a measure motivated by this scenario, referred to as keystroke saving rate (KSR), defined as follows (Trnka and McCoy, 2008):

$$\text{KSR} = \frac{\text{keys}_{\text{normal}} - \text{keys}_{\text{prediction}}}{\text{keys}_{\text{normal}}} \times 100 \qquad (5)$$

where $\text{keys}_{\text{normal}}$ is the number of keystrokes required to type the text without the use of any word suggestions, i.e., it is the number of characters in the text, and $\text{keys}_{\text{prediction}}$ is the number of keystrokes required to type the text assuming that the user always selects a word suggestion if the correct one is available among the n suggestions provided, and that making this selection has a cost of one keystroke. The word suggestions are determined by a language model. In particular, they are the top-n words with highest probability that begin with the prefix of the word that has been typed so far. We consider this evaluation measure for $n = 1, 3, 5$ bearing in mind that $n = 3$ is particularly common for smartphone keyboards. An ideal language model would provide high quality word suggestions, allowing a user to type the text with fewer keystrokes. A higher KSR therefore indicates a better language model.[13]

Although we have described KSR in terms of word suggestions, these can in fact be arbitrary segments of text, as is the case for the language models using SentencePiece. Tokens in SentencePiece incorporate whitespace, whereas this is not part of the tokenization for the word-level language models. For a fair comparison between the two approaches, we include the keystrokes required to enter whitespace when computing $\text{keys}_{\text{normal}}$ and $\text{keys}_{\text{prediction}}$.

In addition to potentially indicating the usefulness of a language model in a word suggestion task, KSR also avoids the challenges of using perplexity to compare word-level language models with those based on open-vocabulary segmentations of the input, such as those produced by SentencePiece.[14]

7.3. Results

In these experiments we use the same training, dev, and test data as in previous experiments. We train SentencePiece models on the training data.

We tuned parameter settings for SentencePiece through preliminary experiments on dev data. We considered byte-pair-encoding (BPE) (Sennrich et al., 2016) and unigram language model (Kudo, 2018) for segmentation. We further explored vocabulary sizes of 1k, 2k, 4k, and 8k. We found BPE segmentation with a vocabulary size of 2k to perform best, and report findings for these settings.

Results are shown in Table 7. Considering first the previous models using word-level tokenization, we see that, for all numbers of suggestions, on both the dev and test data, the model using skip-gram embeddings outperforms that using initialization via the normal distribution, with the exception of $n = 1$ on the dev data. These findings are, overall, consistent with those of the previous evaluations using adjusted perplexity. Moreover, the results for $n = 3$ are perhaps most relevant since smartphone keyboards often provide three suggestions for the next word.

Turning to the results using BPE tokenization, when the normal distribution is used to initialize weights, we see an improvement for both dev and test sets, for each number of suggestions, over both word-level approaches, again with the exception of $n = 1$ on the dev data (although the difference in this case is very small). This finding again indicates the importance of incorporating sub-word information into Mi'kmaq language modelling, in this case through the use of sub-word aware tokenization.

In all but one case, one of the BPE approaches that uses skip-gram to initialize the input layer — either using embeddings trained over a corpus with word-level tokenization or BPE tokenization — gives the best performance. This indicates that there is potential to further improve Mi'kmaq language modelling by initializing embeddings with pre-trained fastText embeddings, even when the language model is trained over BPE segmentation instead of words.

[12] https://github.com/google/sentencepiece

[13] We use KSR to compare approaches to language modelling relative to each other. We therefore do not compare against theoretical upper bounds for KSR as suggested by (Trnka and McCoy, 2008), but intend to do so in future work.

[14] http://sjmielke.com/comparing-perplexities.htm

Tokenization	Initialization	Number of Suggestions (n)					
		$n = 1$		$n = 3$		$n = 5$	
		Dev	Test	Dev	Test	Dev	Test
Word	Normal	1.24	1.09	2.18	2.05	2.89	2.79
Word	Skip-gram	1.06	1.10	2.45	2.44	3.35	3.29
BPE	Normal	1.23	1.17	**2.79**	2.50	3.36	3.44
BPE	Skip-gram (word-level tokenization)	1.19	1.20	2.52	**2.53**	3.65	**3.60**
BPE	Skip-gram (BPE tokenization)	**1.32**	**1.32**	2.62	2.40	**3.81**	3.42

Table 7: Keystroke saving rate for language models trained for both tokenization strategies, with differing approaches to initialization, for varying numbers of suggestions, on both the dev and test sets. The best result on each dataset, for each number of suggestions, is shown in boldface.

8. Conclusions

In this paper we explored a variety of approaches to language modelling for Mi'kmaq, which is particularly challenging due to its rich morphology, and because it is a low-resource language.

We considered n-gram and RNN language models, with a variety of parameter settings in an effort to establish a strong baseline. We then considered the use of pre-trained fastText embeddings to initialize the input layer of the RNN language models. This gave substantial improvements over the baseline, highlighting the importance of sub-word information, and approaches that can represent out-of-vocabulary words, for Mi'kmaq language modelling. We then considered two approaches to language modelling that incorporate cross-lingual word embeddings, but found these to perform relatively poorly. These experiments used adjusted perplexity for evaluation. We then considered an evaluation focused on potential keystroke savings when typing on a smartphone keyboard that offers next word suggestions. In this case we again saw improvements using a language model that incorporated fastText embeddings. Furthermore, we considered language models based on segmentations produced by SentencePiece, specifically using BPE, which include subword units as tokens. We saw further improvements for these models, again highlighting the importance of considering subword units for Mi'kmaq language modelling.

We showed that pre-trained fastText embeddings provided a substantial performance increase for Mi'kmaq language models, and we argued that this is because of the use of sub-word information. The CLWE approaches we considered do not incorporate sub-word information. In future work we intend to explore approaches to learning cross-lingual embeddings at the sub-word level in an attempt to leverage the benefits of sub-word information along with cross-lingual signal from a source language. In our experiments with cross-lingual embeddings, we only considered English as the source language. In future work it would also be interesting to consider morphologically-richer source languages.

Based on the encouraging results for language modelling with SentencePiece segmentation, in future work we intend to further consider language models that operate over sub-word unit tokens, and incorporate morphological analysis (Smit et al., 2014).

Finally, given the small size of the Mi'kmaq corpus, we intend to revisit Mi'kmaq corpus construction, in an effort to build a larger corpus.

9. Bibliographical References

Adams, O., Makarucha, A., Neubig, G., Bird, S., and Cohn, T. (2017). Cross-lingual word embeddings for low-resource language modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 937–947, Valencia, Spain, April. Association for Computational Linguistics.

Battiste, M. (1985). Micmac literacy and cognitive assimilation. In Barbara Burnaby, editor, *Promoting Native Writing Systems in Canada*, pages 7–16. OISE Press/Ontario Institute for Studies in Education, Toronto, Canada.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2014). One billion word benchmark for measuring progress in statistical language modeling. In *15th Annual Conference of the International Speech Communication Association (INTERSPEECH 2014)*, pages 2635–2639, Singapore, September.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

DeBlois, A. D. (1990). *Micmac Texts*. Canadian Museum of Civilazation, Hull, Canada.

DeBlois, A. D. (1996). *Micmac Dictionary*. Canadian Museum of Civilazation, Hull, Canada.

Duong, L., Cohn, T., Bird, S., and Cook, P. (2015). Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 845–850, Beijing, China, July.

Duong, L., Kanayama, H., Ma, T., Bird, S., and Cohn, T. (2016). Learning cross-lingual word embeddings without bilingual corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 1285–1295, Austin-Texas, November. Association for Computational Linguistics.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (PMLR 9)*, volume 9, pages 249–256, Sardinia, Italy, May.

Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool, San Rafael, California.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015)*, pages 1026–1034, Santiago, Chile, December.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Inan, H., Khosravi, K., and Socher, R. (2017). Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling. In *5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, April.

Johnson, P. (1996). Mi'kmaq. In Frederick E. Hoxie, editor, *Encyclopedia of North American Indians*, pages 376–378. Houghton Mifflin Company, Boston, USA.

Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML 2015)*, volume 37, pages 2342–2350, Lille, France, July.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Jurafsky, D., Martin, and James. (2018). *Speech and Language Processing*. Unpublished, third edition. Retrieved April 4, 2019.

Kamholz, D., Pool, J., and Colowick, S. M. (2014). Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 3145–3150, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. (2016). Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 2741–2749, Phoenix, Arizona, February. Association for the Advancement of Artificial Intelligence.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, Detroit, Michigan, USA, May. IEEE.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Maheshwari, A., Bouscarrat, L., and Cook, P. (2018). Towards Language Technology for Mi ' kmaq. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 4139–4143, Miyazaki, Japan. European Language Resources Association (ELRA).

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, April.

Mikolov, T., Karafiat, M., Burger, L., Cernocky, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, pages 1045–1048, Chiba, Japan, September.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Twenty-seventh Conference on Neural Information Processing Systems (NIPS 2013)*, Lake Tahoe, Nevada, USA, December.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013b). Efficient estimation of word representations in vector space. In *International Conference on Learning Representations 2013 (ICLR 2013)*, Scottsdale, Arizona, USA, May.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *Thirty-first Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, California, USA, January.

Press, O. and Wolf, L. (2017). Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Rand, S. T. (1888). *Dictionary of the language of the Micmac Indians : who reside in Nova Scotia, New Brunswick, Prince Edward Island, Cape Breton and Newfoundland.* Nova Scotia Printing Company, Halifax, Canada.

Ruder, S., Vulić, I., and Søgaard, A. (2019). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research.*, 65:569–631.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Smit, P., Virpioja, S., Grönroos, S.-A., and Kurimo, M. (2014). Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden.

Trnka, K. and McCoy, K. (2008). Evaluating word prediction: Framing keystroke savings. In *Proceedings of ACL-08: HLT, Short Papers*, pages 261–264, Columbus, Ohio. Association for Computational Linguistics.

Ueberla, J. (1994). *Analysing a simple language model· some general conclusions for language models for speech recognition*. Ph.D. thesis, Simon Fraser University.