Tagging Scottish Gaelic

Andrew Lee Zupon

10 December 02017

1 Introduction

In this paper, I compare three different methods for part-of-speech tagging Scottish Gaelic, a low-resourced language. I compare a greedy Markov Model tagger, a Markov Model tagger utilizing the Viterbi algorithm, and a Long Short-Term Memory (LSTM) neural network. I show that all of my models achieve higher accuracy than previous work tagging Scottish Gaelic [1].

Part-of-speech tagging is often considered a solved task [2], but low-resourced languages challenge this claim. Much work on part-of-speech tagging is done on well-resourced languages like English, where there is a large amount of training data already available. Corpora for low-resourced languages, if they exist at all, are often smaller than comparable English corpora. Since part-of-speech taggers rely on lots of data for training and testing, this paucity of data adds to the difficulty of creating computational resources for low-resourced languages.

Scottish Gaelic is a Celtic language spoken in Scotland by around 57,000 people. It is considered threatened, but there are ongoing revitalization efforts in Scotland to expand its use. In addition to being a low-resourced language, Scottish Gaelic is also morphologically complex. These two issues make it a challenge for part-of-speech tagging.

Despite the challenges, developing computational resources for low-resource languages is important. First, looking at low-resource languages allows us to test the limits of our current computational techniques and see where there is still room for improvement. Modeling English is useful, but it is only one piece of the linguistic puzzle. Second, creating computational resources for low-resource languages can help the communities that use these languages. Whether a low-resource language is thriving (e.g. Bengali) or at risk (e.g. Scottish Gaelic), creating computational tools allows speakers to use their language in new, expanding domains, which can contribute to the language's survival.

2 Approach

For this project, I use William Lamb's Annotated Reference Corpus of Scottish Gaelic (ARCOSG) [3]. The corpus is divided into eight different domains, four spoken and four written. Each domain contains approximately 10,000 words. The corpus is hand-annotated, and the tagset is based on the tagset for an Irish language corpus. The full tagset includes

246 unique part-of-speech tags, which are marked for a wide variety of complex grammatical and morphological features. There is also a simplified tagset with only 41 unique part-of-speech tags, but the ARCOSG corpus is tagged using the full tagset.

As Manning points out in [2], a lot of the high accuracies obtained for English part-of-speech taggers disappear when tested on domains that the system wasn't trained on. Given eight different domains in ARCOSG, I tried three different approaches for training and testing. Following common practice for English, I first trained and tested on only one domain (News). Next, I trained and tested on all written domains. Finally, I trained and tested on all domains, both spoken and written.

In ARCOSG, each document is separated into clauses, not sentences. For written domains, sentences can be reconstructed by combining lines that don't end in final punctuation, but it becomes more difficult for the spoken domains. In spoken domains, the only final punctuation included is for questions, so reconstructing regular sentences is more difficult. For this reason, for all domains I do not combine clauses into sentences, instead treating each line independently.

To make the training, development, and testing data sets from the full corpus, I combined portions of each relevant domain, making sure not to add the same data to both the training and testing portions. The training set for News (`sg_train_news.txt`) contains 5,288 words, and the testing set for News (`sg_test_news.txt`) contains 4,174 words. The training set for written domains (`sg_train.txt`) contains 18,954 words, the development set for written domains (`sg_dev.txt`) contains 8,705 words, and the testing set for written domains (`sg_test.txt`) contains 11,984 words. For both spoken and written domains, the training set (`sg_train_all.txt`) contains 38,076 words, the development set (`sg_dev_all.txt`) contains 20,086 words, and the testing set (`sg_test_all.txt`) contains 23,789 words. Due to the small amount of data, I also tested the taggers on combined development and testing sets. For just written domains, the large testing set (`sg_bigtest.txt`) contains 20,689 words. For all domains, the large testing set (`sg_bigtest_all.txt`) contains 43,875 words.

My first tagging approach is a greedy Markov Model with add-one smoothing. My second approach is also a Markov Model, but evaluated using the Viterbi algorithm. My third approach is an LSTM neural network built using DyNet. For the LSTM, I preprocessed the data the same way as for the other approaches. I did not find premade word embeddings for Scottish Gaelic, so I randomly initialized embeddings. The LSTM has only 1 layer, a hidden size of 200, and an embedding size of 50. For training the LSTM, I trained it first on 10 epochs and then on 30 epochs.

3 Code

In this section I will discuss some aspects of the code for my Scottish Gaelic taggers.

The greedy Markov Model and the Viterbi taggers are both trained the same way. The training

here involves getting counts from the training data and saving them in new text files. The counts saved during training include the counts of words seen in training (`train_word_counts.txt`), the counts of tags seen in training (`train_tag_counts.txt`), the counts of tag-tag transitions seen in training (`tag_tag_counts.txt`), and the counts of word-tag emissions seen in training (`word_tag_counts.txt`). I also save a list of possible tags for each word seen during training (`observed_wt_pairs.txt`), which plays a role during testing.

During testing, the greedy Markov Model and the Viterbi taggers also start off the same way. After loading in the testing data, I make a list of unknown words and calculate tag transition probabilities from the saved training counts. For the greedy Markov Model tagger, I then loop over the testing data, calculating emission probabilities on the fly. This is where the list of possible tags for known words comes in. For testing, I only consider tags already observed for known words, while unknown words are allowed any potential tag. Doing this may reduce computational complexity, since the system only considers a smaller subset of tags for known words, but at the expense of missing a known word with a novel tag. More testing is needed to determine if this is truly worthwhile.

For the Viterbi tagger, I calculate emission probabilities before looping over the testing data. These emission probabilities do not use the list of possible tags for known words, unlike the greedy Markov Model tagger[1]. After generating the emission probabilities, the system then loops over the testing data, creating the Viterbi trellis. As the results in the following section show, my Viterbi tagger does worse than the basic greedy Markov Model tagger. Furthermore, the greedy Markov Model tagger takes less than 1 minute to test on my hardware, whereas the Viterbi tagger takes over an hour. This is likely due to Viterbi's runtime depending on the number of part-of-speech tags. For the English PTB tagset, which only has around 36 tags, this does not cause too much of a problem. For the full ARCOSG tagset, which has 246 unique tags, this quickly gets out of hand. One potential solution to this runtime issue is to adopt the simplified ARCOSG tagset, which only has 41 tags. Since that requires retagging or modifying the tagged ARCOSG corpus entirely, I set that issue aside for future work.

My LSTM part-of-speech tagger is built using DyNet [4]. The LSTM has one layer and a hidden size of 200. Due to a lack of pretrained Scottish Gaelic word embeddings, I randomly initialize embeddings, which have a dimension of 50. For training and testing I use autobatching, which groups input of the same length together. For the News domain, I use a batch size of 50, due to the smaller amount of data. For written domains and for all domains, I use a batch size of 256. The learning rate is 0.01.

Due to the limitations of the hardware the LSTM was trained on, I only train with two different numbers of epochs: 10 epochs and 30 epochs. As the results in the next section show, 10 epochs gives almost comparable results for written and all domains on overall accuracy, and comparable or higher accu-

[1] Which may contribute to the Viterbi tagger's lower accuracies.

racy for unknown words on all three tests. Training all domains for 30 epochs does yield an advantage overall (accuracy over 80%), but with no real improvement on unknown words.

Due to time and hardware constraints, I did not train the LSTM with 30 epochs for every possible combination of data. Due to the small amount of data I could train the News domain for 30 epochs, and based on my predictions from the greedy Markov Model tagger I chose to train all domains and test with the `sg_bigtest_all` test set. I was unable to train on just the `sg_train_all` training set, since there were part-of-speech tags that only showed up in testing. Because of this, during training I use the tags in both the training and testing data to make tag indices, but the testing tags do not play a role during the actual training. Without this, the matrices do not align, causing errors during testing. Training the larger data sets for 10 epochs takes approximately 1–1.5 hours. Training the same data sets for 30 epochs takes 3–4 hours. Testing takes 1–3 minutes.

4 Results

In this section I will discuss the results of my various Scottish Gaelic part-of-speech taggers on the different training and testing data I used. My results show an improvement over prior work by Lamb and Danso [1], who achieve a maximum accuracy of 76.6% using a Brill bigram tagger.

In 1–2, we see the results for training and testing only on the News domain. These results show an advantage of the greedy Markov Model over Viterbi and the LSTM on overall accuracy, and an advantage of the LSTM on unknown words.

(1)

News - Overall Accuracy	
	`sg_news_test`
MM	**74.17**
Viterbi	63.77
LSTM-10	44.15
LSTM-30	62.68

(2)

News - Unknown Words	
	`sg_news_test`
MM	18.87
Viterbi	15.48
LSTM-10	17.09
LSTM-30	**20.80**

The tables in 3–4 show the results for training and testing on all written domains. Again, these results show the greedy Markov Model beating the Viterbi tagger. In addition, the greedy Markov Model beats the LSTM trained for 10 epochs overall, but the LSTM does better on unknown words.

(3)

Written Domains - Overall Accuracy			
	`sg_dev`	`sg_test`	`sg_bigtest`
MM	**74.17**	**75.00**	**74.50**
Viterbi	64.92	65.70	65.05
LSTM-10	-	-	62.05
LSTM-30	-	-	-

(4)

Written Domains - Unknown Words			
	`sg_dev`	`sg_test`	`sg_bigtest`
MM	18.87	18.31	17.93
Viterbi	14.82	15.56	15.11
LSTM-10	-	-	**19.54**
LSTM-30	-	-	-

In 5–6, we see the results for training and testing on all domains. Again, we see the greedy Markov Model doing better than the tagger with Viterbi. Furthermore, the greedy Markov Model also beats the LSTM neural network tagger run for 10 epochs for overall accuracy. However, when the LSTM is run for 30 epochs, the LSTM overtakes the greedy Markov Model in overall accuracy. Again, even with

only 10 epochs of training, the LSTM beats the greedy Markov Model on unknown words.

(5)

All Domains - Overall Accuracy			
	sg_dev_all	sg_test_all	sg_bigtest_all
MM	78.55	78.07	78.17
Viterbi	71.55	70.97	71.01
LSTM-10	-	-	75.66
LSTM-30	-	-	**80.33**

(6)

All Domains - Unknown Words			
	sg_dev_all	sg_test_all	sg_bigtest_all
MM	17.97	17.47	17.61
Viterbi	14.42	14.88	14.49
LSTM-10	-	-	21.69
LSTM-30	-	-	**22.84**

As these results show, overall accuracies for my three Scottish Gaelic taggers were in the same range of the preliminary results in [1]. The highest accuracy obtained in [1] using a Brill bigram tagger was 76.6%. Testing on News and only on written domains do not reach this accuracy, but two of my taggers do better when trained and tested on all domains. My greedy Markov Model tagger consistently scores higher (>78%) on all testing sets, and my LSTM tagger gets above 80% accuracy when trained for 30 epochs. My results are still preliminary, and I have not tested for statistical significance, but there are clear avenues for future work.

5 Error Analysis

In this section I will discuss some of the error classes I encountered with my Scottish Gaelic part-of-speech taggers. There are three main classes of errors I consider here: (1) words that were tagged incorrectly; (2) gold tags that were not predicted correctly; and (3) predicted tags that were incorrect.

Figure 1 shows the top 10 most incorrectly tagged words. As the figure shows, *an* and *a* are tagged overwhelmingly more incorrectly than the rest of the top 10. Filling out the top ten are other short, common words that could ambiguously have many different tags. For example, *a* alone is seen in the corpus with Ug, Dp3sm, Qq, Sp, and a number of other tags. One potential way to fix these issues may be to simply have more training data.

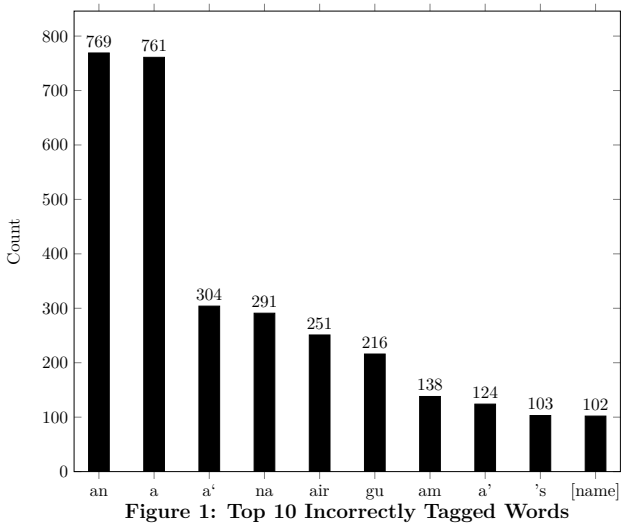

Figure 1: Top 10 Incorrectly Tagged Words

Figure 2 shows the top 10 most frequent gold tags that were not predicted correctly. Most of the top 10 are common noun tags (Nc), with the main difference being the gender (m/f) and case marking

(n/d/g). Given the morphological complexity of the ARCOSG tags, it doesn't seem surprising that this is where some of the errors arise. One potential way to reduce these errors is to use the simplified ARCOSG tagset, but that loses some of the morphological complexity we may be interested in maintaining.

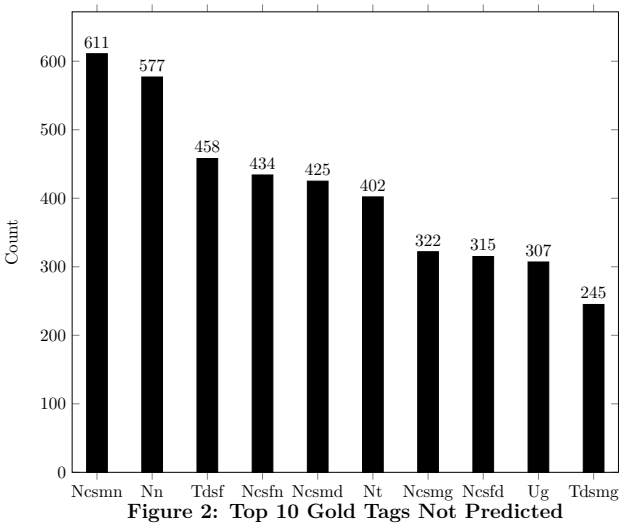

Figure 2: Top 10 Gold Tags Not Predicted

Figure 3 shows the top 10 most frequent incorrect predictions made during testing. By far the most common incorrect prediction is Sp (the basic adposition or preposition tag). The fact that it is incorrectly predicted to such an extent is troubling, especially compared to the other error classes. However, again this may be a reflex of the complexity of the ARCOSG tagset, which has 14 different tags for prepositions alone.

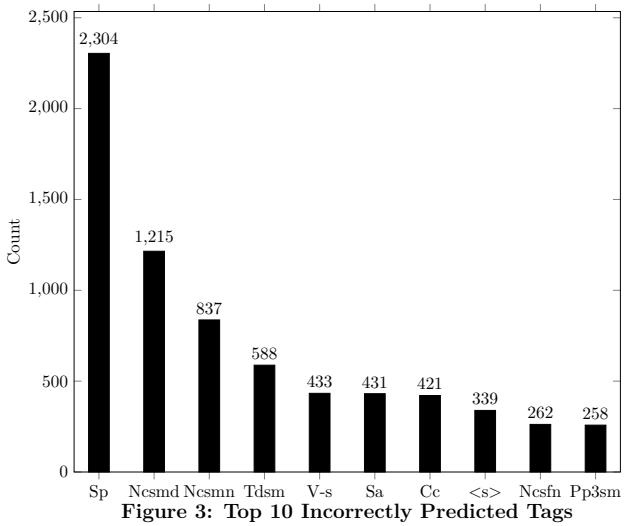

Figure 3: Top 10 Incorrectly Predicted Tags

Looking at these three error classes (words tagged incorrectly, gold tags not predicted correctly, and predicted tags that were incorrect), there are some clear patterns. While my Scottish Gaelic part-of-speech taggers achieve reasonable results, solving these issues is a necessary next step in improving the system. In the following section, I will discuss some ways to approach these error classes, along with other general ideas for improvement.

6 Future Work

The Scottish Gaelic part-of-speech tagging systems presented in this paper are a preliminary work. In this section I will describe a few important ways that this project can be improved and expanded. These improvements can be divided into two main categories: addressing the errors described in the previous section, and improving the robustness of the system overall.

Two of the patterns that emerge by looking at the errors in the previous section are the commonalities

of the types of words to be incorrectly tagged and the types of tags that are not accurately predicted. As mentioned above, a word like *a* shows up in ARCOSG with a wide variety of possible tags. Given this ambiguity, and the relatively small amount of training data, one possible solution is to try to disambiguate these ambiguous words in some way. Since the forms of these incorrectly tagged words don't always change based on their tag, one way to disambiguate may be to implement a trigram tagger. If the immediate context does not provide enough clues, maybe a larger context window could help.

The types of tags that are not accurately predicted follow a similar pattern. Many times, the case and number marking on Scottish Gaelic nouns can overlap with other forms. For example, *cait* 'cat' could be genitive singular, nominative plural, or dative plural. As before, additional context in the form of a trigram tagger may make it easier to predict the correct tag.

Beyond addressing the errors seen in the previous section, there are other ways to improve this project. First, while the results of my part-of-speech taggers are impressionistically better than previous work [1], I have not performed any tests for statistical significance. One clear way to show that my taggers do better than prior work is to show that my accuracy is in fact statistically higher.

Another way to improve this project is by working with the simplified tag set for ARCOSG. As mentioned before, the full tagset used here includes 246 different tags, whereas the simplified tagset only has 41 tags. A lot of the errors described in the previous section appear to stem from closely-related tag possibilities, so simplifying the tagset might improve accuracy in this way. However, switching to the simplified tagset would lose some morphological information, which might cause problems later on.

One limitation of the current study is the hardware and time constraints. In the future, better results might be obtained more easily with more computational resources and more time. As the results in 5 show, the highest accuracy is obtained with an LSTM trained for 30 epochs. For the present paper, it was not feasible to train on more epochs. Given enough computational resources and enough time, however, perhaps a more accurate LSTM can be trained to yield even an even higher accuracy.

Finally, one other important thing that can be done for the future is to simply gather more data. As mentioned at the beginning of this paper, high-resource languages like English have access to lots of training and testing data. For low-resource languages like Scottish Gaelic, that is not always true. Having a corpus of Scottish Gaelic comparable with the English Penn Treebank corpus would make training and testing a part-of-speech tagger for Scottish Gaelic an easier task, and it would allow us to more easily compare the results of part-of-speech tagging crosslinguistically.

7 Related Work

The most relevant related work is Lamb and Danso's Scottish Gaelic part-of-speech tagger [1]. Using a previous version of the ARCOSG corpus, they

built eight different part-of-speech taggers and compared the results. Their best model, a Brill bigram tagger, achieved 76.6% accuracy. Their model also differs from the taggers presented here in that theirs uses a 10% 'hold-out' set for evaluation. They randomly sample 10% of the corpus for evaluation, training on the other 90%.

The main issues observed in [1] also apply to my system. Lamb and Danso note the issue of limited data, stating that the majority of tags occurred fewer than five times in the training set. They also note the importance of morphological information in part-of-speech tagging for Scottish Gaelic, saying that future work could involve integrating some amount of morphological information.

Another related work involves creating word embeddings for Scottish Gaelic [5]. Lamb and Sinclair in [5] discuss the challenge of creating word embeddings for under-resourced languages from sparse data, but their effort is a step towards having real, usable computational resources for Scottish Gaelic. While the LSTM in this paper uses randomly initialized word embeddings, Lamb and Sinclair's work is promising for future work on the language.

8 Conclusion

In this paper I have described three different models for part-of-speech tagging Scottish Gaelic. I have shown that, despite the inherent challenges of working on low-resource languages, my part-of-speech tagger achieves higher accuracies than previous Scottish Gaelic part-of-speech taggers. A basic greedy Markov Model tagger achieves accuracies greater than 78%, and an LSTM trained for 30 epochs achieves an accuracy above 80%.

Despite these achievements, there is always room for improvement. Scottish Gaelic is a morphologically complex language with limited computational resources. Improvements can be made both in the methods of analyzing and tagging different forms, and in improving the amount and quality of Scottish Gaelic data to work with.

References

1. William Lamb and Samuel Danso. (2014). Developing an automatic part-of-speech tagger for Scottish Gaelic. In Proceedings of the First Celtic Language Technology Workshop (pp. 1-5).

2. C.D. Manning. (2011) Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?. In: Gelbukh A.F. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2011. Lecture Notes in Computer Science, vol 6608. Springer, Berlin, Heidelberg

3. William Lamb, Sharon Arbuthnot, Susanna Naismith, and Samuel Danso. (2016). Annotated Reference Corpus of Scottish Gaelic (ARCOSG), 1997-2016 [dataset]. University of Edinburgh. School of Literatures, Languages and Cultures. Celtic and Scottish Studies. http://dx.doi.org/10.7488/ds/1411.

4. Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, and Pengcheng Yin. (2017). DyNet: The Dynamic Neural Network Toolkit.

5. William Lamb and Mark Sinclair. (2016). Developing Word Embedding Models for Scottish Gaelic. PARIS Inalco du 4 au 8 juillet 2016: 31.