Developing a Part-of-speech Tagger for Diplomatically Edited Old Irish Text

Adrian Doyle, John P. McCrae
Insight SFI Centre for Data Analytics, Data Science Institute, University of Galway
adrian.odubhghaill@universityofgalway.ie, john@mccr.ae

Abstract

POS-tagging is typically considered a fundamental text preprocessing task, with a variety of downstream NLP tasks and techniques being dependent on the availability of POS-tagged corpora. As such, POS-taggers are important precursors to further NLP tasks, and their accuracy can impact the potential accuracy of these dependent tasks. While a variety of POS-tagging methods have been developed which work well with modern languages, historical languages present orthographic and editorial challenges which require special attention. The effectiveness of POS-taggers developed for modern languages is reduced when applied to Old Irish, with its comparatively complex orthography and morphology. This paper examines some of the obstacles to POS-tagging Old Irish text, and shows that inconsistencies between extant annotated corpora reduce the quantity of data available for use in training POS-taggers. The development of a multi-layer neural network model for POS-tagging Old Irish text is described, and an experiment is detailed which demonstrates that this model outperforms a variety of off-the-shelf POS-taggers. Moreover, this model sets a new benchmark for POS-tagging diplomatically edited Old Irish text.

Keywords: Old Irish, POS-tagger, Multi-layer, Perceptron, Neural Network, Feature Engineering

1. Introduction

A part-of-speech (POS) tagger adds POS information to individual word and punctuation tokens which comprise a text. POS-taggers are generally employed early in the text preprocessing pipeline, typically being preceded only by tokenisation, though in some cases both tasks are carried out at the same time as a single initial step (Habash and Rambow, 2005). Many downstream NLP tasks, such as automatic term recognition (McCrae and Doyle, 2019) and coreference resolution (Darling et al., 2022), require text to be POS-tagged before they can be applied, and Yocum (2020, 89) claims that the lack of a POS-tagger for Old and Middle Irish has prevented the application of certain authorship attribution techniques to texts from the *Book of Leinster*. Therefore, POS-taggers are extremely important NLP tools which enable the application of a range of follow-on NLP techniques, and the lack of a POS-tagger for Old Irish is already hindering NLP research for the language.

Many types of POS-tagger have been developed over the decades, ranging from simple unigram taggers to complex deep learning models, and Schmid described a multi-layer perceptron (MLP) model for POS-tagging as early as 1994. Many taggers built more recently for a variety of languages still use comparable MLP approaches (Heigold et al., 2016; Hirpassa and Lehal, 2023; Mohammed, 2020; Tesfagergish and Kapočiūtė-Dzikienė, 2020). *The Natural Language Toolkit* (NLTK; Bird et al., 2009) includes several pre-built taggers as off-the-shelf solutions which need only to be trained on text data for a given language. This makes POS-tagging an achievable goal for any language for which training data is available. Generating a sufficient quantity of good quality text data to use for training such models can often be a significant obstacle to the creation of a POS-tagger (Chiche and Yitagesu, 2022, 18), however, particularly for under-resourced languages. This issue takes on another dimension in the case of historical languages like Old Irish, because no more text will ever be created by native speakers than whatever limited quantity has survived from the period in which the language was in use.

For Old Irish in particular several other factors also come into play. Tokenisation, for example, is a non-trivial task for Old Irish (Doyle et al., 2019). The primary reason for this is that words are not consistently separated by spacing in Old Irish. Instead, "... words which are grouped round a single chief stress and have a close syntactic connexion with each other are written as one in the manuscripts" (Thurneysen, 1946, 24). This makes the task of separating tokens difficult, and because tokenisation and POS-tagging are closely related tasks, this also leads to difficulty in POS-tagging.

A considerable amount of lexical variation in Old Irish texts also affects POS-tagging prospects. Old Irish manuscript orthography can be difficult to represent in modern digital editions (Doyle et al., 2018), and different editors represent various orthographic features in different ways. This leads to lexical variation in modern editions which is further increased by the typical spelling variation found in Old Irish manuscripts, and by the morphological complexity of the language. Heigold et al. note that “Morphologically rich languages exhibit large vocabulary sizes and relatively high out-of-vocabulary (OOV) rates on the word level” (2016, 1), which can cause problems for POS-taggers.

Little research to date has focused on POS-tagging for Old Irish, and no work has been published outlining a POS-tagger intended for use with diplomatically edited Old Irish text. The limited amount of work which has focused on Old Irish POS-tagging is discussed in section 3 of this paper. First, however, section 2 discusses the digital corpora which are available for Old Irish, outlining some of the difficulties these corpora create for prospects of developing a POS-tagger. Section 4 gives an overview of several off-the-shelf POS-taggers, before the development of a custom-built MLP model for POS-tagging diplomatically edited Old Irish text is described in section 5. An experiment to measure the accuracies of each of these models is outlined in section 6, and the results of this experiment are discussed in section 7.

Verb	**as·beir**	**do·beir**	**do·gní**
Ind.	*as·beir*	*do·beir*	*do·gní*
	as·biur	*do·beirsem*	*do·gni*
	as·ṁbeir	*do·m-beir*	*do·gníson*
	as·ṁbiursa		
	as·robair		
Dep.	*cenid·epersem*	*ceni·tabair*	*con·déni*
		ní·tabair	*con·deni*
		·tabir	*co·n-déni*
			nád·ṅdéni
			ní·déni
			ní·deni
			ni·deni
			·ṅdéni
			·n-déni

Table 1: Multiple dependent (**Dep.**) and independent (**Ind.**) forms of three Old Irish verbs (*as·beir*, *do·beir*, and *do·gní*) attested in the St. Gall glosses, all of which are analysed as `3sg.pres.ind.` by the St. Gall glosses database (Bauer et al., 2023).

2. Old Irish Text and Corpora

Old Irish refers to the historical stage of the Irish language as it was written from roughly the 7th to the 9th centuries. The majority of Old Irish text which survives in manuscripts dating from this Old Irish period is comprised of three collections of glosses; Würzburg (Wb.), Milan (Ml.), and St. Gall (Sg.). Between the three collections there are about 15,422 glosses written in Irish (Doyle, 2018; e-codices, 2005; Stifter et al., 2021), though these glosses are often very short with many being comprised of only a single word. Aside from these, a small amount of prose, poetry and miscellaneous glosses exist also. As such, the corpus of Old Irish which survives in contemporary sources is not particularly large by comparison to what is available for well resourced, modern languages. Adding to this, a number of factors compound to increase data sparsity within existing digital text repositories for Old Irish.

A considerable amount of code-switching between Old Irish and Latin occurs in each of the collections of glosses. Hence, any POS-tagger for Old Irish would likely be of limited utility if incapable of identifying and tagging Latin text to some extent as well as Old Irish. Spelling is inconsistent within the glosses, and a given word may be spelled multiple distinct ways, even by an individual scribe. The Latin content is variable also, and tends to show “the unusual orthographical peculiarities of Irish manuscripts” (Stokes and Strachan, 1901, xxiii).

Adding to this, Old Irish is morphologically very rich. The verbal complex in particular creates a considerable amount of lexical variability as verbs have both dependent and independent forms, and these forms can change radically in combination with various preverbs, conjunct and emphatic particles, and pronouns (see detailed discussion in McCone, 1997). Depending how verbs are tokenised, this variability can result in many distinct types of token, all representing the same grammatical expression of a single verb (see examples in table 1). Moreover, as an Insular Celtic language, a system of initial mutations can alter the anlaut of words in multiple grammatical situations, and this is expressed in the orthography. For example, the preposition *i* prefixes a nasal, *n*, to the word *degaid*, hence the combination *i ndegaid*. Therefore, both the beginnings and endings of words can change drastically in the orthography of Old Irish.

Further lexical variation is added into the mix by the regular use of abbreviations and contractions in Early Irish manuscripts. Some of these are used to represent set words, morphemes, and letters, such as the Tironian *et* (⁊), used to represent the conjunction *ocus* “and”, ł (Latin *vel*) to represent Irish *nó* “or”, and *·i·*, the Latin symbol representing *id est* (Irish *ed ón*). According to Thurneysen, other abbreviations can be “quite capricious” (1946, 25). Suspension strokes, for example, require a reader to determine from context the missing portion of an abbreviated word, and can therefore represent any number of potential character combinations. These abbreviations and contractions occur in manuscripts alongside the full forms of words in both Latin and Irish. To achieve a high degree of accuracy, therefore, a POS-tagger for Old Irish needs to be capable of tagging both the full forms of words as well as abbreviated and contracted forms.

The process of digitising Old Irish text invariably

Examples	Source	Gloss / Text	Raw Text	Tokens
1(a)	**SGP**	Sg. 1b1	“.i. ci insamlar”	“ci”, “in”, “in·samlar”
1(b)	**CorPH**	Sg. 1b1	“.i. ci in·samlar”	“.i.”, “ci’”, “in·”, “in·samlar”
2	**WBG**	Wb. 9a15	“.i. insamlatharside”	“.i.”, “in”, “samlathar”, “side”
3(a)	**SGP**	Sg. 194a1	“ocondṡruthsin”	“oco”, “nd”, “ṡruth”, “sin”
3(b)	**CorPH**	Sg. 194a1	“ocondṡruthsin”	“oco”, “ond”, “ṡruth”, “sin”
4	**MIDB**	Ml. 2b3	“.i. dintsruth”	“di”, “int”, “sruth”
5(a)	**SGP**	Sg. 7b8	“do·furgabtais”	“do”, “fur”, “-”, “do·furgabtais”
5(b)	**CorPH**	Sg. 7b8	“do·furgabtais”	“do·”, “·fur”, “∅”, “do·furgabtais”
6	**POMIC**	Arm. 64	–	“d-a-beir”, “side”, “0”
7	**WBG**	Wb. 24c16	“daberidsi”	“d”, “a”, “berid”, “si”
8(a)	**SGP**	Sg. 8a8	“da·ṅdichdet”	“d”, “a”, “ṅdi”, “ch”, “da·ṅdichdet”
8(b)	**CorPH**	Sg. 8a8	“da·ṅdichdet”	“d”, “a·”, “·ṅdi”, “ch”, “da·ṅdichdet”

Table 2: Examples of variation in tokenisation style between Early Irish text repositories: **SGP** (Bauer et al., 2023), **WBG** (Doyle, 2018), **MIDB** (Griffith, 2013), **POMIC** (Lash, 2014), **CorPH** (Stifter et al., 2021)

results in further lexical variation between the resulting corpora. Some modern editors aim to produce diplomatic editions, which resemble the text as it appears in the manuscript very closely. Such editors may make use of a large number of Unicode characters in order to represent manuscript features closely, which can result in a more sparse dataset. Other editors may attempt to correct manuscript errors, normalise spelling, supply missing text where manuscripts are damaged or deficient, expand abbreviations and contractions, and introduce ahistorical capitalisation and punctuation. The result is that the same text may be represented differently by two editions (see raw text for examples 1(a) and 1(b) in table 2).

Variation between Old Irish text repositories is even more apparent where tokenisation is applied. All three of the large corpora of glosses have been digitised and lexically annotated, and are available in online (Bauer et al., 2023; Doyle, 2018; Griffith, 2013; Stifter et al., 2021). Two Universal Dependencies (UD) treebanks exist, which contain a small number of POS-tagged and dependency parsed glosses from the Würzburg and St. Gall corpora (Doyle, 2023a,b), and the *Parsed Old and Middle Irish Corpus* (POMIC; Lash, 2014) contains a small amount of POS-tagged Old Irish prose. Each of these text repositories tokenise[1] Old Irish text in different ways, with the result that tokens from one repository are generally incompatible with those of another. Examples 3, 5 and 8 from table 2 demonstrate the same raw text being split into different tokens in accordance with the word-separation methods employed by different repositories[2]. Some repositories also include “empty” tokens representing parts-of-speech which are not realised in the orthography of the raw text (see examples 5(a), 5(b) and 6 in table 2). Finally, certain morphemes, as well as punctuation characters, are repeated in multiple tokens by some repositories, though they appear only once in the raw text (see examples 1, 3(b), 4, 5, and 8 in table 2).

As a result of these varied tokenisation methods, a POS-tagger trained on content from one repository could perform poorly even if tested on the same text content drawn from another repository, because the tokens encountered during training would not be the equivalents of those encountered during testing. This point is almost entirely moot, however, because, of all the repositories listed above, the only ones which share a single style of lexical annotation are the Würzburg glosses (Doyle, 2018) and the two small UD treebanks (Doyle, 2023a,b), all of which use UD-style POS-tags (Zeman, 2016). Aside from these, the only other repository which makes use of an established POS tag-set is POMIC (Lash, 2014), which utilises a variation of Penn-style POS-tags (Santorini, 1990) adapted originally for use with Old English (Santorini, 2016). All of the other text repositories (Bauer et al., 2023; Griffith, 2013; Stifter et al., 2021) use discrete lexical annotations. As such, POS data is not compatible between repositories, with the exception of the UD treebanks.

[1] The terms “token” and “tokenise” are used here in a general sense, referring to the division of text into word-like units which are thereafter annotated. Only Doyle (2018) actually utilises the terms “token” and “tokenisation”, however. Lash (2014) refers to “tokens” only once in POMIC’s annotation manual, but otherwise refers to “words” and “word-division” instead. As such, it would be unreasonable to expect the word divisions of most of these repositories to represent tokenisation in a traditional sense, or to expect tokens from one repository to match those of another.

[2] This point is made only to demonstrate that interoperability between resources is not easily possible. In the context of the methods utilised by individual repositories to divide text, each method is perfectly valid linguistically.

3. Related Work

Only a handful of attempts have been made to develop a POS-tagger for Early Irish. The earliest such attempt was made by Lynn (2012), who describes her model as a “fairly rudimentary” (2012, 23) prototype. Nevertheless, the production of this tagger was impressive as it predated the release of any corpus of lexically annotated Early Irish text. Lynn’s tagger was developed specifically for use with the text, *Táin Bó Fraích* (Meid, 1967), using a manually digitised version of the glossary which accompanied Meid’s print edition as a lexicon. Lynn describes how “The software reads previously unseen text, retrieves part-of-speech information from the machine-readable lexicon for each token in the text and subsequently inserts this information in the text as meta-data” (2012, 22). As the primary aim of Lynn’s work was to demonstrate the value of NLP tools for the field of Early Irish, no results detailing the accuracy of this POS-tagger were published. Presumably, as the lexicon was based on a glossary which had been specifically tailored to the vocabulary of the text used for testing, the tagger would struggle with OOV tokens if applied to unrestricted Old Irish text. Nevertheless, Lynn’s implementation demonstrated at an early stage that, with a sufficiently comprehensive machine-readable lexicon of attested word forms, a POS-tagger for Early Irish may be an achievable goal.

Bauer (2020) has claimed, during a seminar held by the Cardamom project group[3], to have achieved up to 75% accuracy when experimenting with off-the-shelf backoff taggers and Old Irish text drawn ultimately from *Corpus PalaeoHibernicum* (Stifter et al., 2021). Bauer was working with text from the *Annals of Ulster* and the St. Gall glosses. He achieved this 75% accuracy score working only with text from the *Annals of Ulster*, however, when text from St. Gall was included the highest overall accuracy achieved using a backoff tagger was about 30%. A higher overall accuracy of 54% was achieved using a Brill tagger (Brill, 1992). Bauer noted that tokens like preverbs were particularly problematic for tagging. Unfortunately, these results have not been published as of this writing[4].

The next attempt at creating a POS-tagger for Early Irish, and the first to be published in a decade, came when Darling et al. (2022) developed a tagger as a precursor to their work on coreference resolution for Old Irish. This tagger was trained and tested on text from POMIC (Lash, 2014). Normalisation was applied to the text to reduce orthographic variation (2022, 87). Further editing was carried out also, for example, new tokenisation had to be applied where Lash’s word-separation was unsuitable (2022, 87–88), and Lash’s POS-tags were simplified (2022, 88). Darling et al. utilised a Memory-Based Tagger, claiming “it is one of the most effective methods for developing a POS tagger from scratch, since it can learn from such specific features as initial and final characters as well as the context, yielding high rates of accuracy even for extremely small data sets” (2022, 88–89). Darling et al. carried out 10-fold cross-validation to evaluate the tagger, and report a global accuracy of 0.751 when accounting for both seen and unseen words (2022, 89). As texts in POMIC contain ahistorical punctuation, such as hyphenation within the verbal complex, and because Darling et al. had to apply further text normalisation, it is unclear how accurate this model might be if applied to diplomatically edited Old Irish text with more orthographic variation. With one in four words being tagged incorrectly, output from this tagger would still require considerable manual oversight to ensure quality. Nevertheless, these results are impressive given the relatively small amount of data available for training from POMIC. This work, therefore, represents a significant step towards the development of a generally useful tagger for Old Irish, particularly as this was the first such tagger to utilise an established POS tag-set like Penn (Santorini, 1990).

At the time of this writing, no other POS-taggers have been developed for use with Old Irish, and no further attempts have been made to improve POS-tagging prospects. No research has been published to date which addresses the prospect of tagging the type of text which might be found in more diplomatic editions, like *Thesaurus Palaeohibernicus* (Stokes and Strachan, 1901, 1903), and as diplomatic editions like these aim to closely represent Old Irish text as it appears in manuscript sources, this means that no tagger has yet been created which can POS-tag Old Irish as it was actually written. As a POS-tagger is a fundamental NLP tool, this leaves a considerable gap in the list of language resources which are currently available for Old Irish.

4. Baseline Methods

Several types of POS-tagger are available off-the-shelf, and each type may offer different benefits or drawbacks. This section gives an overview of each off-the-shelf model used in the experiment which will be detailed in section 6. As this experiment utilises text from UD treebanks, UDPipe’s bidirectional LSTM POS-tagger (Straka, 2018, 199) is a notable omission from the following list of models used. Unfortunately, no pre-trained UDPipe tagger

[3] `https://cardamom-project.org/`

[4] I would like to express my gratitude to Dr. Bauer for providing me with the relevant slides from his presentation, for discussing his results with me, and for permitting me to reference them here.

currently exists for Old Irish. Moreover, as UDPipe is an entire pipeline for processing `CoNLL-U` files, which includes other steps like tokenisation, a UDPipe tagger could not easily be tested in isolation as is required for this experiment. For these reasons it was not possible to include it in this experiment. The following models are all available through NLTK (Bird et al., 2009).

4.1. Unigram and N-gram Backoff Taggers

Functionally, NLTK's `UnigramTagger` model is the simplest used in this experiment. Bird et al. claim that "Unigram taggers are based on a simple statistical algorithm: for each token, assign the tag that is most likely for that particular token" (2009, 202). Unigram taggers learn specific tokens during training, and therefore, a weakness of these models is that they cannot assign a POS-tag to a token unless that specific token has been encountered during training. This is more problematic for languages like Old Irish, which have a high degree of lexical variation and hence higher OOV rates during testing. Because only the token which is being tagged is taken into consideration during tagging, another limitation of unigram taggers is that the context provided by surrounding words within a sentence is lost, and this can lead to poor results when tagging homographs (Bird et al., 2009, 203).

N-gram taggers, by contrast, can account for the context of a word within a sentence by looking at both the token and the POS-tags of the preceding *n* tokens. This functionality results in a data sparsity problem, however. N-gram taggers must see both a specific token and the preceding *n* POS-tags during training to be able to tag that same combination thereafter. "As *n* gets larger, the specificity of the contexts increases, as does the chance that the data we wish to tag contains contexts that were not present in the training data" (Bird et al., 2009, 205). An n-gram tagger may achieve higher accuracy than a unigram tagger for tokens which it has already seen in specific contexts, but there will be a larger number of tokens which it is incapable of tagging as a result of not having encountered them in particular contexts before. As with unigram taggers, this problem is exacerbated by languages like Old Irish with a high degree of lexical variation.

In order to alleviate the data sparsity issues caused by n-gram taggers, a common solution is to use them in combination with backoff taggers. If an n-gram tagger is unable to identify a POS-tag for a given token, having not seen it in a particular context during training, it will fall back on another POS-tagger model to tag the token instead. It is possible to use multiple layers of backoff taggers, and this is the approach which was used for the experiment detailed in this paper. Any time an n-gram tagger for which $n = x$ could not find a candidate POS-tag for a given token, the model would revert to another n-gram tagger for which the value of $n = x - 1$. This process of falling back on taggers with decreasing n-values would continue until the unigram tagger would finally reached. It was found that beginning with an n-value of $n = 3$ provided the best results.

4.2. Brill Tagger

The Brill tagger (Brill, 1992) is an inductive, transformation-based tagger. According to Bird et al. "Transformational joint classifiers work by creating an initial assignment of labels for the inputs, and then iteratively refining that assignment" (2009, 233). This improves upon n-gram taggers in a couple of ways. Firstly, Brill models can be much smaller than equivalent n-gram tagger models, as they do not need to store large, sparse arrays of n-grams. Secondly, as "The only information an n-gram tagger considers from prior context is tags, even though words themselves might be a useful source of information" (Bird et al., 2009, 208), a Brill tagger can take into account more contextual information. It can account for not only the tag of the preceding token, but also the token itself, and all the same information for the following token.

This functionality requires that the text must first be tagged by a more rudimentary POS-tagger. In the case of the implementation presented here, the unigram tagger described above was used for this purpose. As the Brill tagger trains on this pre-tagged text, instead of storing combinations of tag sequences which have occurred before, it instead develops a set of rules by which it alters certain tags depending on the preceding and following tokens.

4.3. Hidden Markov Model Tagger

Hidden Markov Model (HMM) taggers have comparable benefits to the Brill tagger in that they can take into account a wider range of token contexts than n-gram taggers. HMM taggers "assign scores to all of the possible sequences of part-of-speech tags" (Bird et al., 2009, 233), and then "choose the sequence whose overall score is highest". Like n-gram taggers, HMM taggers take into account both input tokens and the history of predicted tags. Unlike n-gram taggers, however, which use this kind of information to predict the best tag to apply to an individual token in a sequence, HMM taggers generate a probability distribution over tags, then calculate probability scores for sequences of tags by combining these probabilities. The sequence of tags with the highest probability score is chosen. In HMM taggers the HMM is applied in a discriminative manner, not as a generative model.

1.	The **token itself** (buffered, entirely lowercase)	6.	The **last five letters** of the token (all lowercase)
2.	Whether the **token is entirely lowercase** in the sentence (Boolean: true/false)	7.	Whether the **token occurred first** in the sentence (Boolean: true/false)
3.	Whether the **token is entirely capitalised** in the sentence (Boolean: true/false)	8.	Whether the **token occurred last** in the sentence (Boolean: true/false)
4.	Whether the **first letter of the token is capitalised** in the sentence (Boolean: true/false)	9.	The **previous two tokens** (entirely lowercase)
5.	The **first five letters** of the token (all lowercase)	10.	The **following two tokens** (entirely lowercase)

Table 3: Features Collected for Each Token as Input for the MLP Tagger.

4.4. Perceptron Tagger

The perceptron tagger used in this experiment was first implemented by Honnibal and ported over to NLTK from *TextBlob* (2013). It is a neural model which takes various inputs, called features, and uses these to predict the best POS candidate for a given token. According to Honnibal, these features 'will be things like "part of speech at word i-1", "last three letters of word at i+1" etc'. As the model is trained to associate particular features it receives as input with parts-of-speech the weights connecting the various inputs and outputs within the model are increased and decreased in accordance with how useful the model determines they are in aiding it to complete its task. The power of the perceptron tagger to exploit the context of surrounding tokens comes from the features used as input, and the model's own ability to regulate the importance of each of these features as it trains.

5. Methodology

In their review of state-of-the-art POS-tagging solutions, Chiche and Yitagesu concluded that "the use of deep learning (DL) oriented methodologies improves the efficiency and effectiveness of POS tagging in terms of accuracy and reduction in false-positive rate" (2022, 21–22). Several recent papers corroborate this finding, and demonstrate that MLP models often perform well in under-resourced and morphologically rich language settings (Heigold et al., 2016; Hirpassa and Lehal, 2023; Mohammed, 2020; Tesfagergish and Kapočiūtė-Dzikienė, 2020). For this reason a custom MLP tagger was developed for this experiment.

This model differs from the perceptron tagger in a couple of key ways. Firstly, the hidden layers of the MLP tagger should enable it to adapt to non-linearly separable data extracted from the Old Irish text. Secondly, feature engineering for the MLP tagger was customised to focus the attention of the model on aspects of the text which were expected to provide better POS-tagging performance specifically for Old Irish morphology. These aspects were then assessed during ablation analysis to ensure that they did, in fact, provide benefits. For the purpose of feature engineering, ten features were collected from the text for each token (see table 3).

The first feature collected is the token itself. This token is rendered in lowercase to reduce lexical variation, and is then buffered to ensure that all tokens will be of the same length. As the token is rendered entirely in lowercase, features 2 to 4 in table 3 provide information to the model regarding letter case as it is used in the text. Capitalisation does not mark particular parts-of-speech in Old Irish manuscripts, nor hence in diplomatic editions, as it does in modern orthographies, for example, with proper nouns in English or all nouns in German. Capital letters are occasionally employed, however, to match rare manuscript usage of majuscule letters. Majuscule letters are typically employed in manuscripts from this period only at the beginning of paragraphs or significant sections of text, though more than one majuscule letter may be used in sequence. An example of this, drawn from the St. Gall manuscript, can be seen in figure 1, where the initial word of a poem is written entirely using majuscule letters. Given this atypical usage of capitalisation by comparison to modern European orthographies, it was unclear what effect would be produced by either the inclusion or exclusion of features 2, 3 and 4 until ablation analysis was conducted, however, as "POS tagging literature has tonnes of intricate features sensitive to case" (Honnibal, 2013), they were included for this experiment. Their inclusion may also make this POS-tagger more flexible, and better capable of handling less diplomatically edited Old Irish text, where editors employ capitalisation in accordance with modern standards.

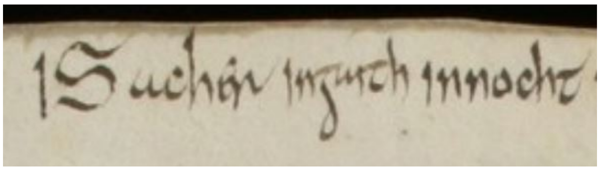

Figure 1: *IS acher ingáith innocht* - St. Gallen, Stiftsbibliothek, Cod. Sang. 904, f. 194 (www.e-codices.ch).

As has been discussed in section 2, both the beginnings and endings of Old Irish words can change drastically in certain grammatical situations. For

	MLP	MLPlus
Hidden layers	3	3
Neurons Per Hidden Layer	64	64
Hidden Layer Activation	ReLU	ReLU
Dropout	20%	20%
Output Layer Activation	softmax	softmax
Training Epochs	50	50
Early Stopping Patience (Epochs)	7	7
Optimiser	Adam	Adam
Rule-based Reassignment Layer	No	Yes

Table 4: Parameters for MLP and MLPlus Taggers.

this reason features 5 and 6 in table 3 focus the attention of the model on the first and last five letters, respectively, of each token. While feature 6 captures morphological information common to many languages, such as case endings for nouns and subject inflections for verbs, feature 5 is intended to cater more specifically to aspects of Old Irish morphology, like initial mutations. Strictly speaking, features 5 and 6 are not individual features themselves, but are comprised of 5 sub-features each. For each token, not only are the first and last five letters collected in combination, but also the first and last four, three, and two letters in combination, as well as the initial and final letters on their own. Therefore, for the word *disruthaigedar*, the following ten sub-features would be collected: *d*, *di*, *dis*, *disr*, *disru*, *r*, *ar*, *dar*, *edar*, and *gedar*. Next, each of these ten sub-features are rendered in lowercase and buffered, like tokens collected for feature 1.

Features 7 to 10 in table 3 relate to the placement of a given token both within the sentence, and relative to other tokens. This kind of information can be helpful in determining POS-tags, as some parts-of-speech are more likely to occur in combination with certain other parts-of-speech. Determiners and adjectives, for example, often occur in combination with nouns, while preverbs and conjunct particles typically precede verbs. Features such as these are not uncommon in POS-taggers, and are also used by Honnibal for his tagger (2013).

Once collected for each token, all ten features were vectorised and one-hot encoded using the `DictVectorizer` class from the `sklearn.feature_extraction` module (Pedregosa et al., 2011). At this point they could be used as input for the model.

Experimentation with hyperparameters during training revealed that the best results were achieved using three hidden layers, with sixty-four neurons per hidden layer. For hidden layers, `ReLU` was used as the activation function, and the `softmax` function was used in the output layer. Optimisation was performed using the `Adam` method (Kingma and Ba, 2015). To avoid overfitting during training, a dropout rate of 20% was used on all hidden layers and early stopping was applied. Validation loss was tracked as a metric to determine when early stopping should occur, and model weights were returned to those which achieved the minimum validation loss during training. An overview of model parameters can be found in table 4.

As the results in section 7 will show, this MLP model performed well relative to other taggers, however, for certain POS-tags which occur particularly infrequently within the corpus, its performance suffered. For this reason the MLPlus model was created. This tagger is almost identical to the first MLP model, except that a rule-based layer is added at the end of the tagging pipeline which reassigns POS-tags for certain tokens. During model training, tokens from the training set which are labeled as interjections, proper nouns, or punctuation are collected. Those which are not homonymous with other tokens which represent more common parts-of-speech are stored in an `infrequent POS-tags` list. When the MLPlus tagger is used during testing it first predicts POS-tags for all tokens, as the MLP model would. Next, a script compares every token in the model's output against each token in the `infrequent POS-tags` list. If a token from the output matches a token in the `infrequent POS-tags` list, the predicted POS-tag is replaced with the POS-tag from the list.

6. The Experiment

6.1. The Data

As has been discussed in section 2, tokenisation methods vary between lexically annotated Old Irish text repositories, and few repositories utilise common POS tag-sets like Penn (Santorini, 1990) and UD (Zeman, 2016). This limits the text available for use in this experiment to either the Old Irish content of POMIC (Lash, 2014), or that of the UD treebanks (Doyle, 2023a,b). Because the text of both of the UD treebanks is diplomatically edited, it was preferable to use UD content in this experiment. It was not possible to also include annotated content from POMIC because this resource separates words differently to the UD treebanks, and utilises a different POS tag-set. This limits the scope of this experiment to diplomatically edited gloss content and a small quantity of poetry. Though it would be preferable to incorporate other genres of text in this experiment, and perhaps text edited to different standards also, the lack of any other corpus which has been tokenised and annotated so as to be compatible with the UD treebanks has ruled out this possibility for now.

The UD corpora are both quite small, with a combined extent of only ninety-eight glosses at the time of this writing. This would not be sufficient to train a POS-tagger, particularly an MLP model. Fortunately, while the master branch of the St. Gall treebank contains only sixty-four glosses at present, the remainder of the corpus has been POS-tagged and annotated with morphological features. This data is stored in the `incomplete.conllu` file which can be found in the development branch[5] of the treebank. Taking into account this content, there are 3,469 POS-tagged glosses containing 21,749 tokens. This should be sufficient to train a reasonably accurate POS-tagger, even on diplomatically edited text. Moreover, Latin tokens in these glosses are are all POS-tagged `X` and annotated with the morphological feature `Foreign=Yes`. This should give taggers the opportunity to learn to distinguish between Latin and Irish text.

6.2. Testing the Models

Because the contents of the St. Gall glosses tend to reflect the thematic context of the Priscian chapter to which they relate, k-fold cross-validation could result in a high number of OOV words unless all glosses within the corpus were shuffled randomly. Instead of randomising all of the data and passing over it sequentially, this experiment uses Monte Carlo cross-validation in order to get a clear picture of each tagger's ability to cope with unseen Old Irish text. This approach required carrying out several passes over the dataset, with each POS-tagger being trained on the same data each pass, then tested on the same test set also.

1,000 passes were carried out in total to ensure the accuracy of the results, while limiting the computational expense of the experiment to a tolerable level. For each pass, 5% of all glosses were split off at random to be used as a test set, and the remainder would serve as the training set. For the MLP and MLPlus taggers, a further 10% of glosses were split from the remainder of the training set at random each pass to be used as a validation set. After all passes for a tagger were complete, the accuracy scores for all passes were averaged to generate the tagger's overall average POS-tagging accuracy. The average accuracy of each tagger over 1,000 passes for each POS-tag can be found in table 5, as well as the total average accuracy for all tokens.

7. Results

As can be seen in table 5, the the unigram and n-gram taggers achieved the lowest scores, 0.698 and 0.708 respectively. The Brill tagger scored marginally better than these, with an accuracy of 0.726. The HMM tagger showed a reasonable improvement over the first three models, with an overall accuracy score of 0.783, and it achieved the highest accuracy scores of any model for tagging determiners and particles specifically. This may speak to the value of calculating probabilities for POS distributions for languages with a lot of lexical variation, over approaches which either rely or fall back on using lookup tables for specific tokens.

The three neural network models offer considerable improvements over all of the other taggers. NLTK's perceptron tagger boasts an 8.5% improvement over the next best performing model, and the MLP model improves upon that by another 2.8%. As has been noted above, the MLP tagger seems to have suffered from under-representation of three particular POS-tags in the data used for this experiment. Only seven tokens were tagged `PUNCT`[6], eighteen were tagged `INTJ` and fifty-four were tagged `PROPN`. The rule-based reassignment layer of the MLPlus tagger seems to have alleviated this issue somewhat as this model achieved the highest accuracy score for `PUNCT`, and showed a marginal improvement for `PROPN`. As these POS-tags represent such a small percentage of the dataset, however, these POS-level improvements do not translate to a significant increase in overall accuracy for the MLPlus tagger. No improvement in overall accuracy can be seen in table 5 as results there are limited to three decimal places. Nevertheless, the MLPlus model is the best performing tagger in most POS categories.

7.1. Ablation Analysis

Ablation analysis carried out on the MLP tagger determined that most of the features outlined in table 3 are beneficial for POS-tagging diplomatically edited Old Irish text, and none hinder the model's performance. It was found that accuracy drops significantly to 0.768 if only the buffered, lowercase token is used as input. Conversely, accuracy remains at 0.896 when features pertaining to letter case (2, 3 and 4 in table 3) are removed from the feature-set. This is to be expected as capitalisation does not mark particular parts-of-speech in Old Irish manuscripts (see discussion in section 5).

Accuracy drops to 0.826 if the feature-set does not include the first and last five letters of each token (features 5 and 6 in table 3), which indicates the value of this morphological information for POS-tagging. Though Honnibal used only the last three letters of tokens as features for his POS-tagger (2013), it was found during experimentation that that capturing up to five letters at the beginning and end of each word produced the best results for the Old Irish text used in this experiment. Using fewer resulted in accuracy drops between 2% and

[5] `https://github.com/UniversalDependencies/UD_Old_Irish-DipSGG/tree/dev/not-to-release`

[6] More punctuation has been included in the latest version of the St. Gall glosses treebank (Doyle, 2023a).

	Unigram	N-gram: n=3	Brill	HMM	Perceptron	MLP	MLPlus
ADJ	0.526	0.530	0.527	0.575	0.694	**0.862**	**0.862**
ADP	0.867	0.825	0.876	0.855	0.893	**0.927**	**0.927**
ADV	0.982	0.982	0.981	0.975	0.974	**0.990**	**0.990**
AUX	0.815	0.831	0.847	0.873	**0.910**	0.896	0.896
CCONJ	0.971	0.966	0.950	0.834	0.956	**0.999**	**0.999**
DET	0.789	0.880	0.886	**0.928**	0.922	0.918	0.918
INTJ	0.656	0.666	**0.678**	0.522	**0.678**	0.000	0.000
NOUN	0.610	0.619	0.612	0.675	0.899	**0.906**	**0.906**
NUM	0.764	**0.790**	0.779	0.703	0.724	0.718	0.718
PART	0.615	0.667	0.775	**0.840**	0.833	0.814	0.814
PRON	0.791	0.747	0.817	0.628	0.814	**0.909**	**0.909**
PROPN	0.121	0.118	**0.124**	0.001	0.055	0.000	0.001
PUNCT	**1.000**	**1.000**	**1.000**	0.415	**1.000**	0.000	**1.000**
SCONJ	0.746	0.790	0.837	0.848	**0.861**	0.832	0.832
VERB	0.532	0.525	0.524	0.776	0.814	**0.880**	**0.880**
X	0.542	0.563	0.566	0.765	0.846	**0.886**	**0.886**
Total Average	0.698	0.708	0.726	0.783	0.868	**0.896**	**0.896**

Table 5: Average POS-tagging Accuracy for all Taggers after 1,000 Training Passes. Best Result per Category in **Bold** and **Underlined**.

7%. This seems to indicate that morphologically significant information for POS-tagging Old Irish penetrates deeper into tokens than is typical of other languages. This can be seen, for example, in the endings of deponent verbs like *suidigidir*, *foilsigidir*, and *cruthaigidir*.

Removing features which inform the model whether a token occurred first or last in a sentence (7 and 8 in table 3) does not appear to affect performance, as the accuracy remains at 0.896. Removing information regarding the following and preceding tokens (features 9 and 10 in table 3), however, drops the accuracy to 0.845.

8. Future Work

Future avenues of research may seek to achieve higher tagging accuracy than the MLP and MLPlus models outlined in this paper by utilising them in combination with other models which require text to be pre-tagged, like the Brill tagger. Though it performed well when tagging punctuation for the dataset used in this experiment, the MLPlus model may be bolstered by supplementing the `infrequent POS-tags` list with a combination of common punctuation characters, and approximations of common manuscript punctuation (such as :⁓, ·~, and .,.,.,) and other symbols (see Groenewegen, 2011). Finally, it is possible that another variety of MLP approach may prove more successful on Old Irish data. Though Heigold et al. found that, for morphologically rich languages, “As long as carefully tuned neural networks of sufficient capacity (e.g., number of hidden layers) are used, the effect of the specific network architecture (e.g., convolutional vs. recurrent) is small for the task under consideration” (2016), more recently Tesfagergish and Kapočiūtė-Dzikienė (2020) have found that a bidirectional LSTM tagger showed notably improved accuracy for Northern-Ethiopic Languages, and Hirpassa and Lehal (2023) found that a variety of bidirectional LSTM tagger performed best for the Amharic Language. It is therefore possible that improvements might be sought over the MLP models presented here by developing a bidirectional LSTM tagger for Old Irish.

9. Conclusion

This paper has described the training of five off-the-shelf POS-taggers, as well as the development and training of two custom-built MLP taggers, on a corpus of diplomatically edited Old Irish text. A comparison of tagging accuracies achieved by these taggers shows that the custom-built MLPlus tagger is the best performing overall, as well as in nine out of sixteen individual POS categories.

A direct comparison cannot be drawn between the scores achieved by taggers used in this experiment and the global accuracy of 0.751 reported by Darling et al. (2022, 89), as each of these experiments utilised not only different corpora of text, but an entirely different POS tag-set. Given the nature of the text data used for this experiment, however, it seems reasonable to suggest that the MLPlus model has set the first benchmark for POS-tagging diplomatically edited Old Irish text.

10. Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Clodagh Downey, whose expertise has been invaluable during the course of this research. Without her guidance and keen attention to detail the current work would not be what it is. Any remaining errors and omissions are entirely my own.

This work has been possible thanks to the support of the Science Foundation Ireland (SFI) as part of Grant Number SFI/12/RC/2289_P2 Insight_2, Insight SFI Centre for Data Analytics. It has also been funded by the University of Galway through the Digital Arts and Humanities Programme, and by the Irish Research Council through the Government of Ireland Postgraduate Scholarship Programme.

11. Bibliographical References

Bernhard Bauer. 2020. ChronHib, CorPH and the Corphusator: Building an Early Irish Corpus. Unpublished.

Bernhard Bauer, Rijcklof Hofman, and Pádraic Moran. 2023. St Gall Priscian Glosses, version 2.1. Accessed: February 12, 2024.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly, Sebastopol.

Eric Brill. 1992. A Simple Rule-Based Part of Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, page 152–155, USA. Association for Computational Linguistics.

Alebachew Chiche and Betselot Yitagesu. 2022. Part of Speech Tagging: a Systematic Review of Deep Learning and Machine Learning Approaches. *Journal of Big Data*, 9.

Mark Darling, Marieke Meelen, and David Willis. 2022. Towards Coreference Resolution for Early Irish. In *Proceedings of the 4th Celtic Language Technology Workshop within LREC2022*, pages 85–93, Marseille, France. European Language Resources Association.

Adrian Doyle. 2018. Würzburg Irish Glosses. Accessed: February 12, 2024.

Adrian Doyle, John P. McCrae, and Clodagh Downey. 2018. Preservation of Original Orthography in the Construction of an Old Irish Corpus. In *Proceedings of the LREC 2018 Workshop: "CCURL2018 – Sustaining Knowledge Diversity in the Digital Age"*, pages 67–70, Miyazaki, Japan.

Adrian Doyle, John P. McCrae, and Clodagh Downey. 2019. A Character-Level LSTM Network Model for Tokenizing the Old Irish text of the Würzburg Glosses on the Pauline Epistles. In *Proceedings of the Celtic Language Technology Workshop*, pages 70–79, Dublin, Ireland. European Association for Machine Translation.

e-codices. 2005. e-codices - Virtual Manuscript Library of Switzerland. Accessed: February 12, 2024.

Aaron Griffith. 2013. A Dictionary of the Old-Irish Glosses. Accessed: February 12, 2024.

Dennis Groenewegen. 2011. Tionscadal na Nod. Accessed: February 21, 2024.

Nizar Habash and Owen Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 573–580, Ann Arbor, Michigan. Association for Computational Linguistics.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2016. Neural Morphological Tagging from Characters for Morphologically Rich Languages. *ArXiv*, abs/1606.06640.

Sintayehu Hirpassa and G.S. Lehal. 2023. Improving part-of-speech Tagging in Amharic Language Using Deep Neural Network. *Heliyon*, 9(7):e17175.

Matthew Honnibal. 2013. A Good Part-of-Speech Tagger in about 200 Lines of Python. Accessed: February 21, 2024.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Elliott Lash. 2014. POMIC Annotation Manual. Manual, The Dublin Institute for Advanced Studies.

Teresa Lynn. 2012. Medieval Irish and Computational Linguistics. *Australian Celtic Journal*, 10:13–27.

Kim McCone. 1997. *The Early Irish Verb*, 2 edition. An Sagart, Maynooth.

John P. McCrae and Adrian Doyle. 2019. Adapting Term Recognition to an Under-Resourced Language: the Case of Irish. In *Proceedings of the Celtic Language Technology Workshop*, pages 48–57, "Dublin, Ireland. European Association for Machine Translation.

Wolfgang Meid, editor. 1967. *Táin Bó Fraích*. The Dublin Institute for Advanced Studies, Dublin.

Siraj Mohammed. 2020. Using Machine Learning to Build POS tagger for Under-resourced Language: The Case of Somali. *International Journal of Information Technology*, 12.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Beatrice Santorini. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision). Standard, Department of Computer and Information Science, University of Pennsylvania.

Beatrice Santorini. 2016. Annotation Manual for the Penn Historical Corpora and the York-Helsinki Corpus of Early English Correspondence. Accessed: February 19, 2024.

Helmut Schmid. 1994. Part-of-Speech Tagging With Neural Networks. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, Kyoto, Japan.

David Stifter, Bernhard Bauer, Elliott Lash, Fangzhe Qiu, Nora White, Siobhán Barrett, Aaron Griffith, Romanas Bulatovas, Ellen Felici, Francesco abd Ganly, Truc Ha Nguyen, and Lars Nooij. 2021. Corpus PalaeoHibernicum (CorPH) v1.0. Accessed: February 12, 2024.

Whitley Stokes and John Strachan, editors. 1901. *Thesaurus Palaeohibernicus*, volume 1. The Dublin Institute for Advanced Studies, Dublin.

Whitley Stokes and John Strachan, editors. 1903. *Thesaurus Palaeohibernicus*, 2 edition, volume 2. The Dublin Institute for Advanced Studies, Dublin.

Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Senait Gebremichael Tesfagergish and Jurgita Kapočiūtė-Dzikienė. 2020. Part-of-Speech Tagging via Deep Neural Networks for Northern-Ethiopic Languages. *Information Technology and Control*, 49(4):482–494.

Rudolf Thurneysen. 1946. *A Grammar of Old Irish*, 2 edition. The Dublin Institute for Advanced Studies, Dublin.

Christopher Guy Yocum. 2020. Text Clustering and Methods in the Book of Leinster. In Elliott Lash, Fangzhe Qiu, and David Stifter, editors, *Morphosyntactic Variation in Medieval Celtic Languages. Corpus-Based Approaches*, pages 85–111. De Gruyter Mouton, Berlin.

Dan Zeman. 2016. UD Guidelines V2. Accessed: February 19, 2024.

12. Language Resource References

Doyle, Adrian. 2023a. *Diplomatic St. Gall Glosses Treebank*. Universal Dependencies. Accessed: February 19, 2024.

Doyle, Adrian. 2023b. *Diplomatic Würzburg Glosses Treebank*. Universal Dependencies. Accessed: February 19, 2024.

Lash, Elliott. 2014. *The Parsed Old and Middle Irish Corpus (POMIC). Version 0.1*. The Dublin Institute for Advanced Studies. Accessed: February 12, 2024.