Leveraging Lexical Resources and Constraint Grammar for Rule-Based Part-of-Speech Tagging in Welsh

Steven Neale[1], Kevin Donnelly[2], Gareth Watkins[1], Dawn Knight[1]
[1]School of English, Communication and Philosophy
Cardiff University, Wales, United Kingdom
{NealeS2, WatkinsG13, KnightD5}@cardiff.ac.uk
[2]Independent Researcher
kevin@dotmon.com

Abstract

As the quantity of annotated language data and the quality of machine learning algorithms have increased over time, statistical part-of-speech (POS) taggers trained over large datasets have become as robust or better than their rule-based counterparts. However, for lesser-resourced languages such as Welsh there is simply not enough accurately annotated data to train a statistical POS tagger. Furthermore, many of the more popular rule-based taggers still require that their rules be inferred from annotated data, which while not as extensive as that required for training a statistical tagger must still be sizeable. In this paper we describe *CyTag*, a rule-based POS tagger for Welsh based on the VISL Constraint Grammar parser. Leveraging lexical information from *Eurfa* (an extensive open-source dictionary for Welsh), we extract lists of possible POS tags for each word token in a running text and then apply various constraints – based on various features of surrounding word tokens – to prune the number of possible tags until the most appropriate tag for a given token can be selected. We explain how this approach is particularly useful in dealing with some of the specific intricacies of Welsh - such as morphological changes and word mutations - and present an evaluation of the performance of the tagger using a manually checked test corpus of 611 Welsh sentences.

Keywords: Part-of-speech, lexical resources, open-source dictionaries, Constraint Grammar

1. Introduction

POS tagging is a well-explored problem in NLP, and highly-accurate taggers have been built using statistical and probabilistic methods for decades. These taggers are typically trained using already-annotated text, from which the probabilities of POS tags being appropriate for certain word tokens are calculated based on features such as the lexical properties of tokens (capitalisation, common prefixes and suffixes etc.) or the POS tags of their *n*-neighbouring tokens. However, the amount of pre-annotated data that these taggers require to be properly trained is considerable, and usually in the region of many hundreds of thousands to millions of word tokens.

For languages such as Welsh – for whom resources are typically more scarce – pre-annotated data in context in these kinds of quantities is very difficult either to create or to obtain. The traditional alternative to probabilistic POS tagging is the rule-based approach, whereby tags are assigned based on pre-defined rules concerning which syntactic categories can be co-located together. Crafting and refining the rules by hand is, however, almost as costly and labour-intensive as producing manually-annotated data from which to train a probabilistic POS tagger.

This paper introduces *CyTag*[1], a rule-based tagger that leverages an open source dictionary and uses the VISL Constraint Grammar parser to assign POS tags to Welsh words in context, using a minimal set of easily-adaptable rules and without the need for millions of tokens of pre-annotated data. Evaluating our tagger using a gold standard dataset consisting of 611 manually checked sentences, we obtain high precision and excellent recall at a level comparable with POS tagging accuracies reported in Welsh and expected in other languages. Our contribution is a robust, open-source and high-performing POS tagger for Welsh – crafted using minimal hard-coded rules – that demonstrates how existing lexical resources can be leveraged to construct accurate taggers for lesser-resourced languages.

2. Background

After decades of development and refinement, probabilistic POS taggers are these days highly accurate – *TreeTagger*[2](Schmid, 1994) tags with a reported accuracy of 94-96%, while the *Stanford Log-linear Part-of-Speech Tagger*[3] (Toutanova et al., 2003) is capable of tagging with an accuracy of over 97%. However, these scores are only attainable after training the tools with considerable quantities of pre-annotated data. Schmid (1994), for example, reports that the accuracy of *TreeTagger* is around 82-84% using a training corpus of 10,000 words, 91-93% with a training corpus of 100,000 words and requires a training corpus of 1 million words to achieve its reported accuracy of 94-96%.

Rule-based POS tagging – whereby pre-defined rules concerning which syntactic categories can be co-located together are used to determine the correct POS tags to assign to word tokens in context – is the traditional alternative to the probabilistic approach. However, this introduces an entirely different bottleneck – considerable time and extensive knowledge is required in order to craft and refine the rules in the first place. The widely-used *Brill Tagger* attempts to

[1] cytag.corcencc.org

[2] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

[3] https://nlp.stanford.edu/software/tagger.shtml

address this bottleneck by automatically acquiring and inferring rules for POS tagging from a running text, and its accuracy has been comparable to that of probabilistic taggers(Brill, 1992). However, more pre-annotated data than is typically available for lesser-resourced languages is still required for this approach, and so again it boils down to a decision (costly either way) between crafting enough rules or annotating sufficient training data by hand.

Examples of POS tagging in Welsh are scarce, but there are some tools available that have attempted to tackle the problem. The *Welsh Natural Language Toolkit (WNLT)*[4] is a bespoke plugin for the *GATE* text processing platform[5] that handles POS tagging for Welsh by looking up terms in a pre-defined lexicon and using hard-coded rules to narrow down ambiguity (words ending in 'iant' or 'cyn' are selected as masculine nouns while those ending in 'es' or 'ell' are selected as feminine nouns, for example) and reports precision, recall and F1 scores of 81%, 82% and 81% respectively. There is also the *Welsh Parts-of-Speech Tagger API*[6], a free-to-use web-based service for tagging Welsh sentences with POS tags and mutations (see section 3.2.1.) developed by the Language Technologies Unit at Bangor University with funding from the Welsh Government.

A far more accurate approach is that adopted by the *Bangor Autoglosser*[7], a multilingual tagger developed to assign POS tags to conversational texts in Welsh, English and Spanish (Donnelly and Deuchar, 2011). The *Autoglosser* is based on Constraint Grammar (CG) (Karlsson, 1990; Karlsson et al., 1995), a language-independent parser whereby easily-adaptable rules based on surface level features and morphology are used to 'discard' ambiguous tags from a list of possible 'readings' for a word token in a running text. By producing a list of possible readings for word tokens from English, Spanish and Welsh dictionaries and pruning those readings with CG-formatted rules, Donnelly and Deuchar (2011) were able to tag the *Siarad* (Welsh-English) and *Patagonia* (Welsh-Spanish) corpora with reported accuracies of 98% and 99% respectively.

3. *CyTag* – A Constraint Grammar-based POS tagger for Welsh

3.1. Motivation – The *CorCenCC* project

Our motivations for developing a bespoke solution for Welsh POS tagging are based on the requirements, aims and scope of the *CorCenCC – National Corpus of Contemporary Welsh (Corpws Cenedlaethol Cymraeg Cyfoes)*[8] project, through which the work is funded. The aim of the project is to construct a 10 million-word corpus of the Welsh language sampled from spoken, written and e-language sources in contemporary contexts, and incorporating crowdsourced contributions to give Welsh speakers the opportunity to involve themselves directly in the creation of the corpus.

3.1.1. The *CorCenCC* POS Tagset

An already completed task for the project has been the development of the *CorCenCC POS Tagset*[9], the current version of which contains 145 fine-grained POS tags collapsing into 13 EAGLES[10]-conformant categories. The necessity for a tagger that is compatible with this tag-set – as well as with the various transcription conventions that have been put in place for marking-up spoken contributions to the corpus – was an influencing factor in our decision to create our own POS tagger rather than to rely on an existing solution. Thus, it is from this bespoke tagset that the POS categories *CyTag* assigns are selected.

Basic	Enriched	Description
E	Ebu	**E**nw **b**enywaidd **u**nigol (noun, feminine, singular)
	Egll	**E**nw **g**wrywaidd **ll**uosol (noun, masculine, plural)
	Epg	**E**nw **p**riod **g**wrywaidd (noun, proper, masculine)
	Epb	**E**nw **p**riod **b**enywaidd (noun, proper, feminine)
	...	
B	Be	**B**erf **e**nw (verb noun, eq. infinitive verb)
	Bpres3u	**B**erf **pres**ennol, **3**ydd pers. **u**nigol (verb, present, 3rd pers. singular)
	Bdyf1ll	**B**erf **dyf**odol, pers. **1**af **ll**uosol (verb, future, 1st pers. plural)
	Bdyf2ll	**B**erf **dyf**odol, **2**il pers. **ll**uosol (verb, future, 2nd pers. plural)
	...	
Rha	Rhaperth	**Rha**genw **perth**ynol (pronoun, relative)
	Rhadib1ll	**Rha**genw **dib**ynnol, pers. **1**af **ll**uosol (pronoun, dependent, 1st pers. plural)
	...	

Table 1: Examples (selected, non-exhaustive) of the relationship between 'basic' and 'enriched' *CorCenCC* POS tags, and their granularity.

We define the EAGLES-conformant categories and the fine-grained POS tags that collapse into them as the 'basic' and 'enriched' tagsets, respectively. Thus, the 'basic' tagset is made up of 13 tags representing major syntactic categories ('noun', 'article', 'preposition', 'conjunction', 'numeral', 'adjective', 'adverb', 'verb', 'pronoun', 'interjection', and 'punctuation') plus two categories representing 'unique' particles to Welsh, and 'other' forms (such as abbreviations, acronyms, symbols, digits etc.). The 'enriched' tagset is the full set of 145 fine-grained tags that collapse down into the 13 categories from the 'basic' tagset, and largely cover different morphological features of the tags in each 'basic' tag category such as gender (masculine or feminine), number (singular or plural), person (1st person,

[4] http://hypermedia.research.southwales.ac.uk/kos/wnlt/

[5] https://gate.ac.uk/

[6] http://techiaith.cymru/api/parts-of-speech-tagger-api/?lang=en

[7] http://bangortalk.org.uk/autoglosser.php

[8] http://www.corcencc.org

[9] http://cytag.corcencc.org/tagset

[10] *Expert Advisory Group on Language Engineering Standards*: http://www.ilc.cnr.it/EAGLES/browse.html

3rd person etc.), or tense (past, present, future etc.). Table 1 shows a selection of nouns, verbs and pronouns from the tagset, demonstrating how the tags themselves are formed consistently from Welsh descriptions of their morphological features (always ‘ll’ for plural forms, always ‘b’ for feminine and ‘g’ for masculine, etc.).

3.2. The *CyTag* process

Following a similar methodology to the one behind the *Bangor Autoglosser* (Donnelly and Deuchar, 2011), *CyTag* is based on Constraint Grammar (CG) (Karlsson, 1990; Karlsson et al., 1995), and in particular is built around the latest version of the software – VISL CG-3[11]. As described in Section 2., CG works by applying rules that ‘discard’ ambiguity by ‘pruning’ a ‘cohort’ (list) of the available ‘readings’ (possible tags) for a given word token, based on the surface and morphological features of either itself or its neighbouring word tokens. Although this means that we must be able to produce a list of possible readings (POS tags, morphological information etc.) for each token in a running text, these readings do not have to be in context – it is the job of the CG rules to choose the correct reading based on context later. Thus, unlike most POS tagging solutions, the initial cohort of readings can be extracted from word-level lexica such as dictionaries, which even in lesser-resourced languages are far more readily-available than fully annotated sentences in context.

Thus, *CyTag* assigns POS tags to tokens using three steps:

- A list of possible POS tags is produced for each token,
- Using CG-formatted rules, the list of possible tags for each token is pruned to as few as possible (ideally one),
- The optimal tag for each token is selected, using additional processing to help select a tag in any cases that were still ambiguous after running CG.

3.2.1. Producing possible POS tags for each token

An initial list of possible POS tags for each token is produced over two steps. Firstly, any tokens whose definite POS can be identified without needing to look it up elsewhere are assigned the appropriate tag outright – for example, regular expressions are used to determine whether or not the token is a punctuation mark (‘.’, ‘!’, ‘?’), a digit (‘1980’), or a symbol (‘£’, ‘%’). Next, gazetteers are checked to determine whether or not the token is already known to be an acronym – such as ‘GIG’ (‘NHS’ in English) – or an abbreviation – such as ‘Cyf.’ (‘Ltd.’ in English) or ‘e.e.’ (‘e.g.’ in English).

Once all of the tokens with definite POS tags have been identified as such, the second step is to look up the remaining tokens in a pre-defined lexicon, currently containing lemmas and POS tags for 210,438 Welsh word forms. The lexicon has been extracted from *Eurfa (v3.0)*[12], the largest Welsh dictionary available under an open license and containing approximately 211,000 word forms derived from 10393 Welsh lemmas. As well as lexical categories such as nouns, verbs, and pronouns, *Eurfa* contains full morphological information for each entry – including gender, person (first, second, or third), number (singular or plural) and tense (present, past, imperfect etc.). This allows entries to be easily mapped to tags from the CorCenCC POS tagset for inclusion in the pre-defined lexicon.

After all of the possible entries for a given token have been found in the lexicon, a CG-formatted cohort of readings (list of possible POS tags for the token) can be produced. For example, given the token ‘a’ we get the following readings:

```
“<a>”
        “a” {9,18} [cy] Cys cid :and:
        “a” {9,18} [cy] Rha perth :who:
        “a” {9,18} [cy] U gof ::
```

Looking up the token in the lexicon has told us that there are three possible readings for ‘a’, which is the 18th token in the 9th sentence and is in Welsh (‘[cy]’) – firstly, it could be a coordinating conjunction (‘Cyscid’) equivalent to ‘and’ in English; secondly, it could be a relative pronoun (‘Rhaperth’) equivalent to ‘who’ in English; and thirdly, it could be an interrogative particle (‘Ugof’), which is often used at the start of a clause in which a question is asked and has no real equivalent in English.

One particular nuance of Welsh that we begin to deal with at this stage of the *CyTag* process is mutation, a common phenomenon in Welsh in which the first letter of a word can change (or ‘mutate’) depending on, for example, the preceding word or on word order – some examples of how this phenomenon works can be seen in Table 2. We deal with mutation by first checking if the token begins with a known mutated form – along with a code denoting mutation type (‘am’ for aspirate, ‘nm’ for nasal, ‘sm’ for soft, or ‘hm’ for added ‘h’) – and then adding what would be it’s de-mutated form to a list of possible tokens to look up in the lexicon. For example, *CyTag* would catch the ‘ch’ (aspirate mutation of the letter ‘c’) at the start of ‘char’ or the ‘ngh’ (nasal mutation of ‘c’) at the start of ‘nghar’ and would add ‘car’ (also ‘car’ in English) to the list of possible de-mutated forms for either ‘car’ or ‘nghar’. To handle one particular case of a soft mutation in which the letter ‘g’ is dropped from the beginning of a word, we remove the letter ‘g’ from the start of every token we encounter that starts with it and add the now ‘g’-less form to the list of possible de-mutated tokens.

If after these checks the list of possible de-mutated forms for a given token is populated, we look up each of these possible forms in the lexicon and if they are found, we add them to the cohort of readings for the token. Thus, when the readings are pruned in the second step of the *CyTag* process, any possible de-mutated forms can be taken into account by the CG-formatted rules, depending on which mutation type would have caused the change in form and on the (selected or potential) readings for preceding tokens. Given the token ‘mae’, for example, the original search of the lexicon will return a reading for the present tense, third-person singular verb (‘Bpres3u’) ‘bod’, equivalent

[11] http://visl.sdu.dk/cg3.html

[12] For more information, see: http://www.eurfa.org.uk/

Mutation	Effect	Example
aspirate (am)	c → ch p → ph ...	car → ei char (her car) pensil → ei phensil (her pencil)
nasal (nm)	c → ngh b → m ...	car → fy nghar (my car) bag → fy mag (my bag)
soft (sm)	ll → l rh → r ...	llyfr → ei lyfr (his book) rhosyn → ei rosyn (his rose)
added ‘h’ (hm)	a → ha w → hw ...	afal → ein hafal (our apple) ysgol → ein hysgol (our school)

Table 2: Examples (selected, non-exhaustive) of the four major mutation types in Welsh.

to ‘is’ from ‘to be’ in English. However, thanks to our mutation lookup rules – which know that any word word beginning with the letter ‘m’ could be the nasally-mutated form of a word beginning with the letter ‘b’ – we can also account for the possibility that ‘mae’ is a nasally-mutated (‘nm’) form of the masculine singular noun (‘Egu’) ‘bae’, equivalent to ‘bay’ in English:

```
“<mae>”
        “bod” {4,19} [cy] B pres 3 u :be:
        “bae” {4,19} [cy] E g u :bay: + nm
```

A second important nuance of Welsh that it is important to us account for when producing cohorts of readings from the lexicon is the common occurrence of elision in Welsh, which is complicated by the fact that a) shortened (or ‘elided’) forms of words can become attached either to the start *or* the end of surrounding words, and that b) the same elision could have potentially come from more than one full word. For example, an ‘f’ elided onto the beginning or an ‘m’ onto the end of a word could both have originally been the word ‘fy’; an ‘n’ elided onto the end of a word could have originally been either of the words ‘yn’ and ‘ein’; an ‘r’ elided onto the end of a word could have originally been either of the words ‘y’ and ‘yr’. We deal with elisions by running a multiple lookup of the all the possible words that a particular elision could have originally been, and returning the full list of readings for those possibilities – for example, given an ‘n’ elided onto the end of a word as our token, we return readings for three possible forms of ‘yn’ – an uninflected preposition (‘Arsym’) corresponding to the English ‘in’, a predicative particle (‘Utra’), and a verbal particle (‘Uberf’) – as well as a reading for ‘ein’ – a first-person plural dependent pronoun (‘Rhadib1ll’) corresponding to the English ‘our’:

```
“<’n>”
        “yn” {8,2} [cy] Ar sym :in:
        “yn” {8,2} [cy] U tra ::
        “yn” {8,2} [cy] U berf ::
        “ein” {8,2} [cy] Rha dib 1 ll :our:
```

3.2.2. Pruning the list of possible POS tags for each token

Once the cohort of possible readings for each token has been constructed, the next step in the *CyTag* process is to pass the list to VISL CG-3, which has been provided with the path to a bespoke ‘grammar’ file containing rules to help ‘constrain’ or ‘prune’ the readings for those tokens that are still ambiguous. The grammar currently contains 243 rules, each designed to select or remove certain options from the cohort of readings for a given token depending on the POS, morphological features and/or mutation type of its neighbouring tokens. Rules are formatted as follows:

action (reading) if (neighbour (features))

whereby *action* is what the rule should do (select the reading, remove the reading etc.), *reading* is the particular reading for a given token that the action should be performed on, *neighbour* is the neighbouring token of interest on whose features the action depends (1 for the following token, -1 for the preceding token etc.) and *features* are the POS, morphological features and/or mutation type of the neighbouring token that we expect to find in order to satisfy the rule and perform the original action on the reading.

An example in practice is the disambiguation of the token ‘yn’, which is often used in conjunction with the word ‘mae’ (from the verb ‘bod’, ‘to be’ in English) either side of a noun as a connecting particle – it can either be a predicative particle linking the noun to other nouns and adjectives, or a verbal particle linking it to another verb. For example, if we consider the short phrase:

“Mae Cymru (hefyd Saesneg: Wales) yn wlad Geltaidd”

translating to the English:

“Wales (English: Wales) is a Celtic country”

or as a very literal token-to-token translation:

“(be) Wales ... (*‘yn’ particle*) Celtic country”

For this segment, an initial reading could look as follows:

```
“<Mae>”
    “bod” {377,1} [cy] B pres 3 u :be:
“<Cymru>”
    “Cymru” {377,2} [cy] E p b :Wales:
...
“<yn>”
    “yn” {377,9} [cy] U tra ::
    “yn” {377,9} [cy] U berf ::
    “yn” {377,9} [cy] Ar sym :in:
```

```
“<wlad>”
    “gwlad” {377,10} [cy] E b u :country: +sm
“<Geltaidd>”
    “Celtaidd” {377,11} [cy] Ans cad u :Celtic: +sm
```

Here, we want our CG grammar to be able to select for us that in this context the token ‘yn’ is the predicative particle (‘Utra’) linking the nouns ‘Cymru’ (‘Wales’) and ‘wlad’ (‘country’), and not the verbal particle (‘Uberf’) or the preposition (‘Arsym’) corresponding to the English word ‘in’. Assuming no previous rules have decided that the token *should* be tagged as a verbal particle or as a preposition, the grammar should eventually arrive at the following rule:

```
SELECT (“yn” U tra) if (1 (E));
```

Here, we instruct the grammar to *select* the reading where the token “yn” correspondes to the predicative particle (‘Utra’) *if* the following token (‘1’) is a noun (‘E’). Because we already know that the ‘yn’ in the example phrase is followed by a noun (‘wlad’, a soft mutation of ‘gwlad’ – note the ‘+sm’ in the readings), the verbal particle and the preposition are discarded and the predicative particle can be selected.

In the previous example, we can see how words that could have been mutated are represented in the cohort of readings. Another example in practice shows how the presence of possible mutations can be used by the grammar to select the appropriate reading for a given token. For example, if we consider the short phrase:

> “cwmnïau cydweithredol yng Nghymru...”

translating to the English:

> “cooperative companies in Wales...”

For this segment, the initial reading looks as follows:

```
“<cwmnïau>”
    “cwmni” {58,16} [cy] E g ll :companies:
“<cwdweithredol>”
    “cydweithredol” {58,17} [cy] Ans cad u ...
“<yng>”
    “yn” {58,18} [cy] Ar sym :in:
    “fy” {58,18} [cy] Rha dib 1 u :my:
“<Nghymru>”
    “Cymru” {58,19} [cy] E b u :Wales: + nm
```

Here, we can see that the word ‘Nghymru’ is a nasal mutation (‘+ nm’) of the word ‘Cymru’ (‘Wales’ in English), and knowing that such mutations occur after prepositions we can implement the following rule in the CG grammar:

```
SELECT (Ar) IF (1 (nm));
```

Here, the grammar will *select* the reading corresponding to a preposition (‘Ar’) *if* the following token (‘1’) has been affected by a nasal mutation (‘nm’), ensuring that the preposition corresponding to the English word ‘in’ is correctly selected for the token ‘yn’ in this context, as opposed to the pronoun (‘Rhadib1u’) corresponding to the English word ‘my’.

3.2.3. Post-CG disambiguation steps

Once the CG grammar has pruned the cohorts to as close to one reading per token as possible, some final steps are employed to try and eliminate any remaining ambiguity. The simplest of these is that when a token has two readings that have the same POS tag – but different meanings in English, hence two readings – then the POS tag that the readings share is selected. For example, the Welsh word ‘ceisio’ and its soft mutated form ‘geisio’ both produce two readings, corresponding to their two English meanings in the lexicon extracted from Eurfa (‘try’ and ‘seek’):

```
“<ceisio>”
        “ceisio” {178,17} [cy] B e :try:
        “ceisio” {178,17} [cy] B e :seek:
```

However, as both of these readings have the same POS tag – ‘Be’, an infinitive verb (also known in Welsh as a ‘verb noun’) – we can safely assign this tag to the word token in the running text.

A similar process is used when a word token is encountered that has been pruned to two readings of proper noun, but it has not been possible to discern the gender of the token. For example, when producing the initial cohorts of readings, any word that is not in our lexicon but begins with a capital letter – such as ‘Eleanor’ – might be assumed to be a proper noun. Because we cannot deduce the gender of the proper noun at this stage, a cohort of two readings (‘Epg’ for a masculine and ‘Epb’ for a feminine proper noun) is produced:

```
“<Eleanor>”
        “Eleanor” {16,3} [cy] E p g :Eleanor:
        “Eleanor” {16,3} [cy] E p b :Eleanor:
```

Word tokens such as these are searched for in a small collection of *gazetteers* – lists of stand-alone terms which have been leveraged from linked open data by running simple SPARQL queries against DBpedia[13]. These gazetteers contain lists of:

- Masculine given names,
- Feminine given names,
- Surnames,
- Place names.

If the word token can be found in either of the masculine or feminine given name gazetteers, then the appropriate tag (‘Epg’ for masculine or ‘Epb’ for feminine) can be selected. If the word token is found in both given name gazetteers, in the surnames gazetteer, or in the place names gazetter, then the morphological information about gender is discarded and the token is tagged as a ‘neutral’ proper noun (‘Ep’).

[13] See: `http://wiki.dbpedia.org/` or `http://dbpedia.org/sparql` for the SPARQL endpoint

Or, if the word token is not found in any of the gazetteers, then the token is tagged as a 'neutral' proper noun.

The most complex post-CG disambiguation steps involve the querying of two bespoke dictionaries, which are produced based on the tags found in a 611 sentence gold standard evaluation corpus (see Section 4.). These dictionaries comprise a *tag-token coverage dictionary*, and a *tag-sequence dictionary*.

Tag-Token Coverage Dictionary: This dictionary is created by taking each individual word token in the input (611 sentence gold-standard) corpus, and counting the number of times that token is assigned each CorCenCC POS tag. The final dictionary contains the most commonly-assigned tag for each unique word token. For example, the word 'yn' – which as we know from the first example in section 3.2.2. can either be a predicative particle ('Utra'), a verbal particle ('Uberf'), or a preposition ('Arsym') corresponding to the English word 'in' – is most commonly tagged as a preposition in the input corpus, and thus represented in the tag-token coverage dictionary as:

{"yn": "Arsym"}

If we encounter a word token that is still ambiguous at this point, we can check whether it is present in the tag-token coverage dictionary, and if it is we can then assign it the POS tag that would most commonly be given to that token.

Tag-Sequence Dictionary: This dictionary is created by cycling through every 3 token n-gram in a given sentence in the input (611 sentence gold-standard) corpus, and recording the POS tags of the tokens either side of the middle token, which we replace with the word 'find'. The 3-gram of POS tags either side of the word 'find' is then stored in the dictionary with the the word that 'find' replaced. For example, upon finding a 3-gram in the corpus with POS tags of 'Bpres3u' (present tense verb, 3rd person singular), 'Egu' (masculine singular noun), and 'Utra' (predicative particle) – a common 3-gram combination which would be produced by a simple phrase such as 'mae [noun] yn...' or '[noun] is' in English – the following entry would be added to the tag-sequence dictionary:

{"['Bpres3u', 'find', 'Utra']": "Egu"}

If we find that a word token is still ambiguous after all of the preceding disambiguation steps, we can now query the dictionary to see if it contains a 3-gram of 'find' surrounded by the POS tags of the n^{-1} and n^{1} word tokens. Continuing with the example 3-gram above, if the ambiguous word token was preceded by 'Bpres3u' and followed by 'Utra', then we could consider 'Egu' as the POS tag to assign to the ambiguous token.

4. Evaluation

We have evaluated the performance of *CyTag* using a 611 sentence (14,876 token) gold standard evaluation corpus that has being constructed as part of the ongoing work on the *CorCenCC* project. The corpus is comprised of eight example input files (included with the *CyTag* software) containing excerpts from a variety of existing Welsh corpora – *Kynulliad3*[14] (Welsh Assembly proceedings), *Meddalwedd*[15] (translations of software instructions), *Kwici*[16] (Welsh Wikipedia articles), and *LER-BIML*[17] (multi-domain spoken corpora) – and from the short abstracts of three additional Welsh Wikipedia articles. The 611 sentences were first tagged using *CyTag*, and then each token in the resulting output was manually checked by a Welsh language speaker. If an incorrectly tagged token was found, the correct POS (in line with the *CorCenCC POS Tagset* outlined in Section 3.1.1.) was noted down instead.

	No. of tokens
Total	14,876
Pre-CG:	
– with only one reading	8,917
– with multiple readings	5,198
– with no readings	761
— assumed to be proper nouns	504
Post-CG:	
– disambiguated	14,403
— pruned to one reading by CG	13,461
— two readings with same POS	65
— ambiguous gender proper nouns	504
— found in gazetteer	1
— tag from coverage dictionary given	372
– still ambiguous	216
– unknown	257

Table 3: Token counts at various stages of the *CyTag* process.

Table 3 shows how many tokens have been disambiguated at different stages of the *CyTag* process. From a total of 14,876 tokens in the 611 sentence input corpus, a total of 14,115 tokens have been assigned readings from Eurfa prior to CG being run. Of these, 8,916 have been assigned a single reading, with 5,198 having been assigned multiple readings that will need to be disambiguated. Of the 761 tokens that were not assigned a reading, 504 tokens have been assumed to be proper nouns (due to their capitalisation) and will have been automatically assigned two readings each – masculine proper noun ('Epg') and feminine proper noun ('Epb') – leaving 257 tokens unknown. After CG has been run, 14,403 tokens have been disambiguated (pruned down to one token), leaving the 257 tokens that were unknown prior to CG being run, and 216 tokens that are still ambiguous – CG and our various post-CG disambiguation steps were unable to prune these tokens down to a single read-

[14] http://cymraeg.org.uk/kynulliad3

[15] http://techiaith.cymru/corpws/Moses/Meddalwedd

[16] http://cymraeg.org.uk/kwici

[17] http://www.lancaster.ac.uk/fass/projects/biml

ing. Of the 14,403 tokens that were disambiguated, 13,361 of these were pruned by the CG rules, 65 of them were tokens that had been assigned two readings pre-CG but with the same POS tag (on account of two different meanings in English), 504 of them were proper nouns with ambiguous gender, 372 of them were assigned the most likely tag based on their presence in the coverage dictionary (see Section 3.2.3.), and 1 token was ambiguous, but found in the *CorCenCC* gazetteers described in Section 3.2.3..

	POS Type	
	Basic tags	**Enriched tags**
Tokens	14,876	
Tagged	14,403	
Still ambiguous	216	
Unknown	257	
Tagged correctly	13,866	13,488
Precision	96.27	93.64
Recall	96.61	96.52
F1	96.44	95.06

Table 4: Results of running *CyTag* over the 611 sentence corpus, taking into account performance over both the basic and enriched sections of the *CorCenCC POS Tagset*.

Table 4 shows the results of running the 611 sentences through *CyTag*, taking into account the difference in performance if we consider the full, enriched section of the *CorCenCC POS Tagset* or only the basic categories section into which the enriched tags collapse. The table demonstrates that from a total of 14,876 tokens, *CyTag* was able to assign a POS tag to 14,403 of them, with 216 tokens still being ambiguous post-CG and 257 tokens left unknown. Comparing the output of *CyTag* to the same 611 manually checked sentences from the gold standard evaluation corpus, we can see that 13,866 tokens have been assigned the correct tag from the basic POS categories, while 13,488 tokens have been assigned the correct tag from the enriched POS categories – this results in precision, recall and F1 values of 96.27, 96.61 and 96.44 over the basic POS categories and 93.64, 96.52 and 95.06 over the enriched POS categories, respectively.

	POS Type	
	Basic tags	**Enriched tags**
Tokens (multiple readings)	5,198	
Tagged correctly	4,885	4,800
Precision	93.70	91.99
Recall	95.31	95.23
F1	94.50	93.58

Table 5: Results of running *CyTag* over the 611 sentence corpus, considering its performance over only those tokens that had multiple readings pre-CG.

Table 5 shows the results of running the 611 sentences through *CyTag* when only those tags that were assigned multiple readings prior to CG. Comparing the output of *CyTag* to the same 611 manually checked sentences from the gold standard evaluation corpus, we can see from the table that from a total of 5,198 tokens *CyTag* was able to assign the correct basic POS tag to 4,885 tokens, and the correct enriched POS tag to 4,800 tokens. This results in precision, recall and F1 values of 93.70, 95.31 and 94.50 over the basic POS categories and 91.99, 95.23 and 93.58 over the enriched POS categories, respectively.

Finally, Table 6 shows how successfully each component of *CyTag* was able to assign POS tags, considering how accurately readings were pruned to one by CG and how accurate each of the post-CG disambiguation steps described in Section 3.2.3. proved to be. As the table demonstrates, CG is able to prune correctly prune readings for a given token to just one with a high degree of accuracy (97.33% for basic POS tags and 94.88% for enriched POS tags), and our disambiguation step of stripping one of the readings out when a token has two readings with the same tag (on account of the token having different meanings in English represented in Eurfa) is also highly accurate. Only one ambiguous token was found in the *CorCenCC* gazetteers, and so although it was correctly tagged we cannot discern whether other words might be tagged erroneously after being found in the gazetteers. We have had reasonable success in assuming that capitalised words that were unknown prior to running CG were proper nouns – 80.56% of these assumptions turned out to be correct, although the accuracy for these tokens using the enriched tagset is rather lower (71.23%) due to the fact that in many of these cases it was simply not possible to discern the gender of the proper noun. However, the accuracy of our tagging of ambiguous tokens based on their presence in the coverage dictionary (see Section 3.2.3.) is lower at 41.13%.

	POS Type	
	Basic tags	**Enriched tags**
Pruned to one by CG	13,101 (97.33%)	12,772 (94.88%)
Two readings, same tag	65 (100%)	64 (98.46%)
Proper noun assigned	406 (80.56%)	359 (71.23%)
Found in gazetteer	1 (100%)	1 (100%)
Found in coverage dict.	153 (41.13%)	153 (41.13%)

Table 6: Success rate of pruning readings to one using CG, and the various post-CG disambiguation methods also employed.

5. Discussion

The results of running the 611 sentences through *CyTag* are very positive – F1 scores above 95% represent a marked improvement over *WNLT* and are approaching the reported accuracy of the *Bangor Autoglosser*. The strong performance of *CyTag* is also notable in the context of our gold standard corpus and the diverse range of sources from which the 611 sentences are extracted – Giesbrecht and Evert (2009) observe that reported POS tagging accuracies in the high 90%s usually come from focused evaluations on refined or edited texts with few errors or non-standard forms, while true accuracies over unseen texts from diverse sources are more likely to fall to the low 90%s or even the high 80%s. We can therefore be confident that *CyTag*'s performance would be consistently high across many different domains and sources – vital in the context of tagging the balanced, representative National Corpus of Contemporary Welsh for the *CorCenCC* project.

There are a number of particularly encouraging observations that can be made about the results presented in Section 4.. As Table 5 demonstrates, *CyTag* also performs well – with F1 scores approaching 95% – on only those tokens which had more than one reading prior to CG being run. Thus, while it is true that a large number of tokens are disambiguated by default as a result of only having one possible reading to begin with, the overall results are not necessarily skewed by this, and our CG-formatted rules are able to prune the more ambiguous tokens with good effect. This is highlighted by the breakdown of how accurate various disambiguation steps are, as demonstrated in Table 6: the accuracy with which we can prune the readings for a token down to one using CG is clearly highlighted, and this is supported by the accuracy with which certain assumptions – that unknown, capitalised words are probably proper nouns, and that two readings with the same tag can be cut down to one – can be applied.

Also noteworthy is the fact that in all of our evaluations, the results for enriched POS tags are very close to those obtained when considering only the basic POS tag categories, despite there being 145 enriched tags compared to only 13 basic tags, and thus much more room for error with the enriched set. Were it the case that our results were far better over the basic POS tagset than over the enriched tagset, we could conclude that the tagger was able to determine the major categories of words, but was having more trouble identifying morphological features (such as noun gender or number) and exceptions. Thankfully, this is not the case, and it's much more likely that only a small-number of cross-category discrepancies need to resolved in order to yield improved results. For example, we notice that *CyTag* can have trouble determining which POS tag to assign to the versatile token 'yn', with a number of instances where it has been tagged as a predicative particle ('Utra') where it should have been an uninflected preposition ('Arsym'), or vice-versa.

Moving forwards, implementing new rules to address these kinds of cross-category discrepancies should be more straightforward than if we were required to try and address inter-category discrepancies such as ambiguous noun genders or verb tenses, and enable us to prune readings for a given token with increased accuracy. This would mean less tokens remaining – or less readings for a remaining token – at the post-CG disambiguation stage, where again there is room for further development. In particular, our coverage dictionary could be redesigned or improved, any additional techniques for discerning the gender of proper nouns explored, and new disambiguation steps explored and developed, in order to handle more of those tokens that cannot be pruned by CG-formatted rules alone and to further increase the standard to which we can POS tag Welsh sentences in context.

6. Conclusions

We have described *CyTag*, a rule-based tagger for Welsh that leverages lexical information from an open source dictionary and uses Constraint Grammar to select the most appropriate POS tags for words in context. The high precision and excellent recall of the tagger – as demonstrated by our evaluation over a gold standard dataset of 611 manually-checked Welsh sentences – are very promising for Welsh and are in line with the accuracy expected of POS taggers over unseen text in other languages. As well as a high-performing open-source POS tagger for Welsh, our work demonstrates that by leveraging existing knowledge and resources and with minimal, easily-adaptable rules, accurate taggers can be developed even for languages for whom pre-annotated training data is scarce.

In future work, we intend to build additional layers of auxiliary tagging into *CyTag* to account for cases of function being different to pure form – for example, nouns being used as adjectives or plural forms being used as honorific singular forms (such as 'chi', a third-person plural pronoun which is often used to address people respectfully in the singular). We will also focus on the recognition and tagging of multi-word expressions (MWEs), with Welsh having a number of word and token combinations that make little sense outside of the multi-word context (such as 'ar agor', which can be treated as a single adjective equivalent to 'open' in English). Finally, in the context of the *CorCenCC* project, we will be using *CyTag* as the foundation of an extended pipeline incorporating a Welsh adaptation of the *UCREL Semantic Analysis System (USAS)*[18](Piao et al., 2018), in order to assign both syntactic and semantic tags to the National Corpus of Contemporary Welsh.

Acknowledgements

This work has been funded by the UK Economic and Social Research Council (ESRC) and Arts and Humanities Research Council (AHRC) as part of the *Corpws Cenedlaethol Cymraeg Cyfoes (The National Corpus of Contemporary Welsh): A community driven approach to linguistic corpus construction* project (Grant Number ES/M011348/1).

References

Brill, E. (1992). A Simple Rule-based Part of Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing (ANLC '92)*, pages 152–155. Association for Computational Linguistics.

Donnelly, K. and Deuchar, M. (2011). Using constraint grammar in the Bangor Autoglosser to disambiguate multilingual spoken text. In *Constraint Grammar Applications: Proceedings of the NODALIDA 2011 Workshop*.

[18] `http://ucrel.lancs.ac.uk/usas/`

Giesbrecht, E. and Evert, S. (2009). Part-of-speech tagging – a solved task? An evaluation of POS taggers for the Web as corpus. In *Proceedings of the 5th Web as Corpus Workshop (WAC5)*.

Karlsson, F., Voutilainen, A., Heikkilä, J., and Anttila, A. (1995). *Constraint Grammar: A Language-Independent Framework for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin / New York.

Karlsson, F. (1990). Constraint Grammar as a Framework for Parsing Running Text. In *Proceedings of the 13th Conference on Computational Linguistics (COLING '90)*. Association for Computational Linguistics.

Piao, S., Rayson, P., Knight, D., and Watkins, G. (2018). Towards a Welsh Semantic Annotation System. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC 2018)*.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Toutanova, K., Klein, D., Manning, C., and Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pages 252–259. Association for Computational Linguistics.