Medieval Irish and Computational Linguistics*

TERESA LYNN
Macquarie University & Dublin City University

THIS paper will consider the application of some NLP (Natural Language Processing) techniques to Medieval Irish texts to provide an alternative perspective on linguistic analyses of such texts. Using *Táin Bó Fraích* as a case study, I present the outcome of some preliminary experiments. The pilot study starts with the creation of an annotated lexicon as a basis of automated text analysis. Linguistic features such as part of speech information are recorded in a machine-readable representation to assist with subsequent linguistic analysis of this well-studied text. Using *CELT*'s electronic version of Meid's 1974 edition, I conduct both statistical and linguistic analyses of textual features such as sentence structure, lexical frequency and grammatical types. I use the results of this analysis to raise some tentative suggestions regarding *Táin Bó Fraích*, and in particular the frequently noted relationship between the two distinct sections. On this basis I hope to make some suggestions about the potential usefulness of applying some NLP techniques to Medieval Irish.

1 PROJECT DESCRIPTION

1.1 Introduction

Computational Linguistics is an interdisciplinary field of study, which combines the application of computational methods with those of linguistic processing. Computational methods describe the way in which computer science techniques are used to solve problems. Linguistic processing refers to the many ways in which human languages are analysed, understood and used. The combination of both of these fields arose from attempts in the

* I thank the anonymous reviewer of this paper for valuable comments and feedback.

1950s to make machines (computers) understand language in the same way as humans do. Computers have since been designed to 'model' the activities of the human brain that are related to language processing—both in written text and speech.

While computers will never fully substitute the linguistic abilities of the human brain, much progress has been made in this field to date. In many ways, we unknowingly benefit from computational linguistic research. Everyday tools such as spell-checkers, thesauri, grammar-checkers and hyphenators make our lives easier when dealing with written language. Online translation systems are useful when it comes to 'getting the gist' of a foreign text, the content of which may not be accessible to us otherwise. Search engines, such as Google or Bing, have changed the way we access information in today's world. Mobile telephone SMS applications, voice-activation, and automated response systems have all been developed off the back of NLP research.

While it is clear that many of the world's most successful corporations have benefited from this work, how does it help us at a research level? The answer lies with the ability of computers to automate many of the manual tasks we need to undertake as part of our work. If our research involves trawling through documents and analysing text, then why not take advantage of today's computerised world to make that task easier and quicker?

This paper aims to draw attention to some of the options available to those involved with researching texts. While it is understandable that many who have chosen a career in historical or linguistic studies may not have extensive technical experience, an awareness of these tools may open doors to collaborative efforts across disciplines.

1.2 Application to Linguistic Research

As Jurafsky & Martin (2009, 37–38) point out, engaging in complex language behaviour (human or computer-based), requires various kinds of knowledge of language; phonetics and phonology, morphology, syntax, semantics, pragmatics and discourse. Computational linguists therefore apply their knowledge of these linguistic features to the development of language tools. Some of the tools which are of interest to those working in the field of linguistic research are:

—corpus analysis tools (which analyse bodies of text);

—machine translation systems (for researching other languages);

—automated parsing tools (to retrieve syntactical information from the text);

—language generation tools (to assess the accuracy of linguistic grammar theories).

There are certain tasks that the human brain cannot cope with. In linguistics, this is often the case when a large amount of data is being analysed. The computational capacity of computers, on the other hand, renders these tasks more manageable. Take for instance a basic statistical analysis of language use over time. If we wanted to manually analyse the use of the terms 'citizens', 'democracy', 'freedom', 'duties' and 'America' throughout the history of US presidential inaugural speeches, we would have a mammoth task of collecting the texts, manually scanning for instances of these terms (while facing the risk of human error) and calculating the trends across time. Instead, linguistic computer-based tools can provide a platform that makes this work less time-consuming and possibly more accurate. The *Natural Language Toolkit* (*NLTK* = Bird & al., 2009a) provides a good example of the analysis of the use of these five words between 1785 and 2005; see Figure 1. This dispersion plot highlights the change in trends of word usage over time. The analysed corpus was created by joining the texts of the Inaugural Address Corpus end-to-end. In the dispersion plot, 'word offset' denotes the position of the word within the file. Each stripe therefore represents the location of each occurrence of the word in question within the corpus. This type of corpus analysis can prove useful and interesting as it not only reveals linguistic information but it also may reveal some social changes over time.

2 Application to a Medieval Irish Text

Of the various applications available within the NLP domain, I will focus on the tools available for corpus analysis. In particular I will discuss the various corpus analysis techniques made

available by *NLTK*. This kit provides open-source programming modules, linguistic data and documentation. 'Open-source' means that while the book and tools have been developed by three main authors, it is now open to the public as a collaborative project. It is constantly being updated and revised, in particular as the tools are applied to new languages. The open-source nature of the kit means all the necessary resources are also available online (Bird & al., 2009b).

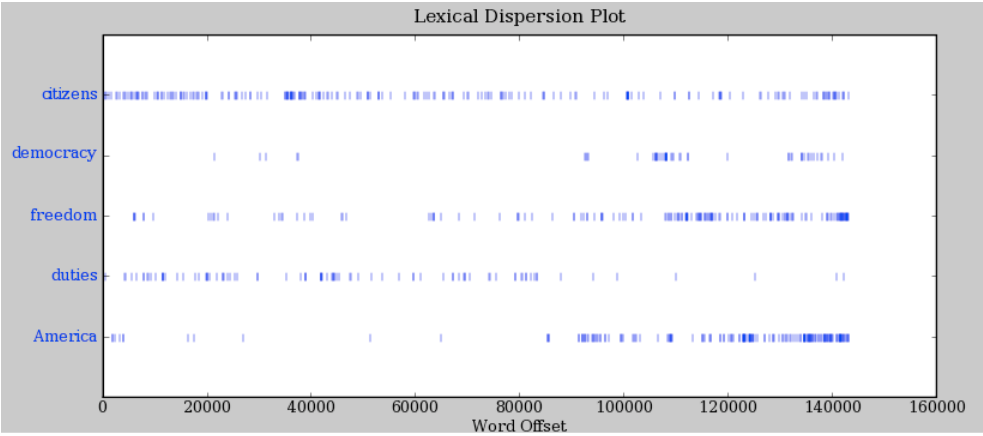

Figure 1: Dispersion plot showing the use of several terms over time

For the purpose of this experiment, I have chosen *Táin Bó Fraích* (*TBF*), an early medieval Irish text recorded in several early Irish manuscripts and based on a saga of the warrior Fróech. The tale is short in comparison to others of its kind and it is divided into two parts. Part one (*TBF*1) deals with Fróech's wooing of Findabair and the difficult dealings he has with her parents. (Her mother is the legendary Queen Medb.) Part two (*TBF*2) is a much smaller part and it deals with the *táin* ('cattle raid') that he takes part in following a deal he makes with Medb and her husband Ailill.

There are several different transcribed versions of this text available.[1] I have chosen Meid's 1974 edition for this task. The text is available in digital format edition from the *CELT* (Corpus of Electronic Texts) website (Färber 2001). For linguistic processing purposes I have adapted this version to a sentence-

[1] The latest one known to me is Meid 2009.

per-line format, with sentences being delimited by periods, exclamation marks and question marks.

The analyses performed below are not so significant in the results they provide but rather in the demonstration of the types of computational techniques that can be applied to this text. The important thing to note is that, while *TBF* was chosen as a manageable size text for a small project, most of these techniques can similarly be applied to any machine-readable texts.[2]

Firstly, I used the *NLTK* tokenizer (Chapter 3) to separate each token in the text with a space. The reason for this is to separate punctuation from words. Otherwise, strings of text such as {*máthair*.} and {*máthair*} would be regarded by a computer as different tokens. Once the text was tokenized, it was possible to perform interesting linguistic analyses on the language used in the text.

2.1 Lexical Dispersion Plot

Similarly to the example in Figure 1, I have used *NLTK* to create a dispersion plot depicting the appearance of or reference to some of the characters throughout *TBF*: *Findabair*, *Medb, Ailill* and *Conall*. See Figure 2. This provides an almost 'birds-eye view' of the characters' presence in the text. At a glance we can see that *Medb* and *Ailill* appear quite frequently together, although Ailill is mentioned by name a little more often throughout the tale. We can also see that the characters Findabair and Medb only appear in the first part of the tale, *TBF*1, while Conall appears only in the second part of the tale, *TBF*2.

2.2 Frequency Analysis

Tokenised text provides a solid basis for frequency analyses. Such tasks are often time-consuming when undertaken manually. However, with *NLTK* and *Python* programming language these tasks are much easier and results are available in a fraction of the time.

[2] 'Machine-readable' does not apply to scanned PDF files, which are mere images of texts. However, such images can often be converted to machine-readable text with the use of OCR (Optical Character Recognition) software.

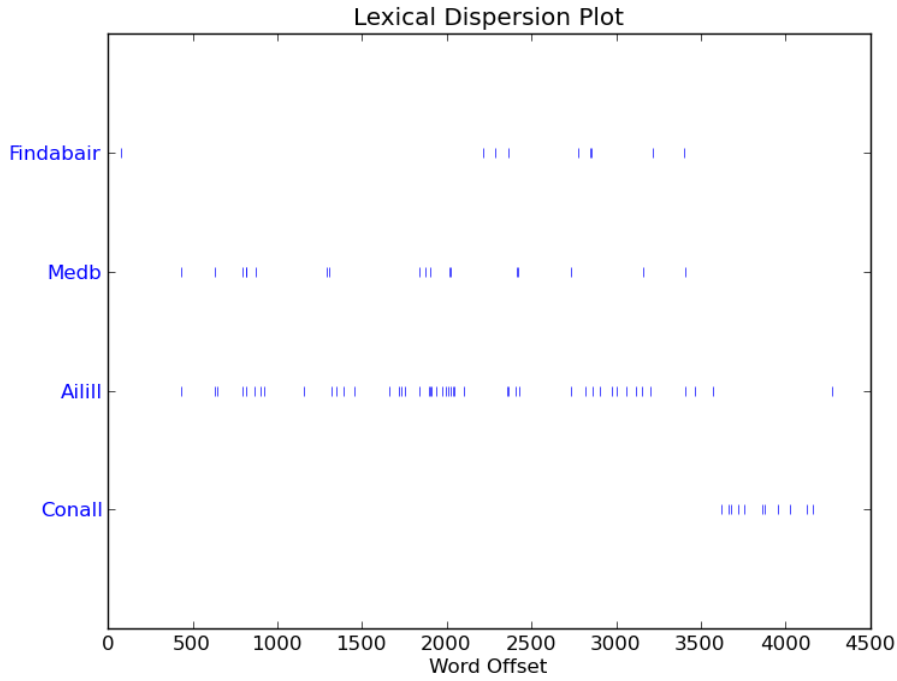

Figure 2: Dispersion plot showing the characters throughout *Táin Bó Fraích*

For this task I divided *TBF* into its two separate parts in order to compare the parts of the story through frequency analysis. Figure 3 provides some lexical analysis results on the text of *TBF*1 and *TBF*2. The number of tokens represents the number of words and symbols present in the text. In contrast, the number of types tells us only about the number of distinct (unique) words in the text. For example, the word *sí* 'she' appears frequently in the text. A calculation of the number of tokens would tell us how many times *sí* occurs. A type count, on the other hand, would return a value of 1 to indicate that the unique token *sí* simply exists in the text. The number of types will always be lower than the number of tokens as human language texts generally display reuse of vocabulary to some degree. With this information we can then calculate the lexical diversity of the texts. This is a calculation of the number of types proportionate to the number of tokens. The result indicates the richness of vocabulary in each text. While the results do not differ tremendously, they are in fact significant as it has been observed by some readers that the language in the second part of the tale appears much simpler than that of the first part. The results may also reflect the fact that, as Meid (1974, xi;

cp. 2009, 19) points out, part two is 'very poorly narrated' in comparison to part one.

2.3 Word Length Analysis

In Figure 3, we see the average word length for each part of the *TBF* text. What these results did not tell us is the range of word lengths throughout the text. A quick calculation with *NLTK* provides the graph in Figure 4, which shows us the variations in these counts. The X-axis shows the word lengths (the longest word is 14 characters long) and the Y-axis shows the count of words for each word length. We can see from this, for example, that roughly 400 words in *TBF*1 are 3 characters in length.

	TBF 1	TBF 2
# tokens	3773	899
# types	1372	400
lexical diversity	2.75	2.2475
ave. word length	4	3
ave. sent. length	12	10

Figure 3: Results of Frequency analysis on *TBF*1 and *TBF*2

Note that while the high percentage of words of length 1 in *TBF*1 may be reflective of the frequency of punctuation marks, a separate calculation of word length averages on punctuation-free text returned substantially the same results as the calculation on the text inclusive of punctuation.

2.4 Concordance

A more traditional corpus analysis technique is concordance analysis. A concordancer shows the context in which a word or string of characters may occur in a corpus. This type of analysis can tell us a lot about word patterns within a language and is often used by linguists and language teachers. *NLTK* provides a concordance module that enables the user to build a concordance of a given text.

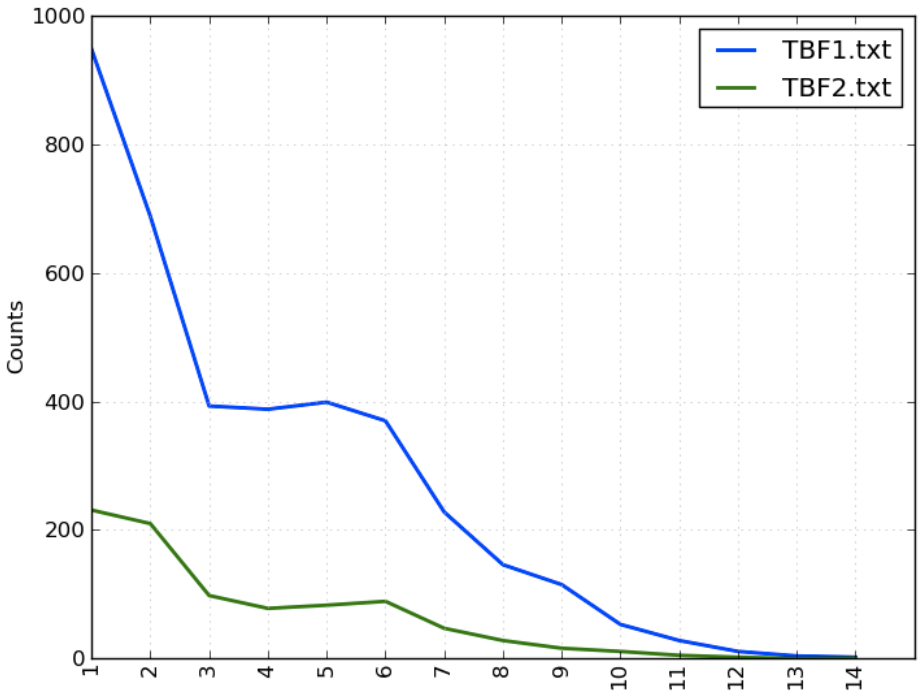

Figure 4: Counts for word length in *TBF*1 and *TBF*2

In this instance, the input I provided was the entire tokenised *TBF* text. For experimental purposes, I chose the character *Medb* to see the context in which her name would occur. Within seconds, the program produced a concordance output, which can be seen in Figure 5. (Note that I performed four analyses in total in order to include data for the inflected forms of the word *Medb*.) The width of the output has been restricted to 75 characters per line. This ensures the neatly aligned display we see and also explains broken tokens at each side in some cases.

The results can be analysed from various viewpoints, depending on the user; linguist, historical linguist, historian, literary historian etc. Two patterns appear to jump out at first glance. Medb usually only appears in the tale alongside her husband, Ailill. In fact, they speak in unison on more than one occasion. But more interesting is the choice, in only two instances, to list the pair as Medb & Ailill instead of Ailill & Medb.

2.5 Machine-Readable Lexicon

A machine-readable lexicon is a valuable resource to have when working on any language. If a lexicon is available in machine-readable format, the computational techniques that can be

performed on a language become more extensive. Such a lexicon can contain not only glosses for each word, but also meta-data such as part-of-speech information, alternative spellings, verb valency, gender, number etc. For the purposes of building a prototype in this instance, it was not feasible to create a lexicon to cover medieval Irish in general. Such a task would be time consuming and also require the skills and input of an expert linguist of this language. Instead, I created a machine-readable lexicon whose coverage would extend to the *TBF* text alone.

```
on dún inna lín .\nÓ gabais Ailill & Medb flaith , nícos tánic riam & nícos ti
nai .\n" Fo chen dóib ", ol Ailill & Medb .\n" Is óclách án fil and ", ol Aili
riu .\n" Fo chen dúib ", ol Ailill & Medb .\n" Iss ed doróachtamar ", ol Fróec
\n" Níba turas ar aig thaig ón ", ol Medb .\nImbrid Medb & Ailill fidchell íar
ar aig thaig ón ", ol Medb .\nImbrid Medb & Ailill fidchell íar sin .\nGaibid
ll .\n" Ní hed is accobor limm ", ol Medb , " acht dul do imbirt na fidchille
ss ed laithe inso as síam limm ", ol Medb , " Deithbir ón ", ol Fráech .\n" At
eora aidchi and ."\nLa sodain atraig Medb .\nBa mmebul lée buith donaib ócaib
rum .\nImmosnacaillet íarum Ailill & Medb .\n" Farbbiba sochaide n - immund de
\nbine fornn ."\n" Is líach ón ", ol Medb , "& is meth n - einich dúnn ."\n" N
ht arandálfarsa ."\nDothóet Ailill & Medb issa rrígthech .\n" Tíagam ass ", ol
agéuin Ailill íarum .\n" Tairchi , a Medb ! " , ol Ailill .\nDothóet Medb íaru
 , a Medb ! " , ol Ailill .\nDothóet Medb íarum .\n" In n - aithchéin sin ? "
ib i tírib Connacht .\nTéit Ailill & Medb ina ndún íarum .\n" Mórgním doringén
íarum .\n" Mórgním doringénsam ", ol Medb .\n" Issinn aithrech ", ol Ailill ,
omun aile thíssad .\nAtraig Ailill & Medb & dogníat aithrigi ndó dond écht dor
nd éicni anúas .\nDosféccai Ailill & Medb .\n" Dalei co ndercar ", ar Fráech ,
air .\n" Arotnaisc dó ", ol Ailill & Medb , "& tair chucunni cot búaib do thái

nCarthai Findabair , ingen Ailella & Medba , ara irscélaib .\nAdfíadar dósom oc
ora ochtga humai for imdai Ailella & Medba immdernide de chrédumu uili is sí i
amad a thabairt dó ar omun Ailella & Medba .\nÍar sin gataid Findabair
a hétach
ride ar búaib sceo mnáib dosoifet la Meidb & Ailill .\nAtbélat fir la clúaiss n
ch Froích .\nÍar sin adgládar Fróech Meidb .\n" Is maith ro ngabus fritt ", ol
untir .\nCelebraid íarum do Ailill & Meidb .\nDocumlát dá críchaib íarum .\nEcm
h & form anmain airec co Ailill & co Meidb com búaib do tháin na mbáu a Cúalngi
 & a baí laiss , co luid la Ailill & Meidb do tháin na mbó a Cúanlngiu .\nFinit

hrelam , ní thibrind i tindscra cid Meidbi insin ."\nDoching úadaib asa taig í
```

Figure 5: Concordance for *Medb* in the text of *TBF*

I based my lexicon on the vocabulary available in Meid's edition of *TBF*. Fortunately, the vocabulary is rich in information and each head entry also lists the alternative or inflected forms that appear in the text. However, the structure of the data in Meid's vocabulary does not lend itself easily to computational processing. (Of course, this vocabulary section was not created with computer processing in mind.) For this reason, a considerable amount of time was spent tidying up the data into a

consistent representation for the lexicon. Despite this extensive work, it is important to note that once the lexicon is built, further linguistic analyses which employ the lexicon are relatively quick.

Figure 6 is an extract from *TBF*'s machine-readable lexicon. As it shows, each lexical entry contains four slots of meta-data, delimited by commas. The first holds the headword, the second holds the part-of-speech information, the third holds a list of glosses and the fourth holds a list of spelling variations. By separating the information in a structured way, it is easily retrievable with computational techniques. Within the lexicon alone, some interesting linguistic information can be seen. For example, by using a text editor such as Emacs[3] we can perform a search on a part of speech and view the highlighted results quickly to observe the density of different types of part-of-speech tags in the lexicon for *TBF*. It is also easier for scholars or learners of the language to retrieve gloss information from the lexicon with a digital search function.

```
ascid, noun, [gift], [ascedaib]
as-indet, verb, [tells, narrates], [asndíth, aisnid]
aisndís, verbal_noun, [telling], []
ass, noun, [milk], [aiss]
as-luí, verb, [escapes, elopes], [as-élub, as-éláfa]
```

Figure 6: Extract from the *TBF* machine-readable lexicon

2.6 Part-of-Speech Tagged Corpus

With the machine-readable lexicon at our disposal, it was then possible to build a rudimentary part-of-speech (pos)-tagger. This tool was developed using Python programming language. 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. This marked-up text is referred to as a pos-tagged corpus. This tool was then applied to the *TBF* corpus to produce a pos-tagged corpus—see Figure 7. This type of corpus reveals interesting information, not only about the text but also about syntactical patterns within a language. Such a resource is not

[3] Emacs is an open-source text editor available at <*http://www.gnu.org/software/emacs/*> [accessed on 2 September 2011].

only valuable to linguists but would also be highly beneficial to learners of the language. For example, the additional information available in a pos-tagged corpus can lend itself to more interesting concordance searches where part-of-speech patterns can be queried alongside tokens.

```
Atagéuin (verb) Ailill (Prop) íarum (adverb) .
" (punct) Tairchi (verb) , (punct) a (pron) Medb (Prop) ! (punct) " (punct) ,
(punct) ol (verb) Ailill (Prop) .
Dotháet (verb) Medb (Prop) íarum (adverb).
" (punct) In (int_particle) n - aithchéin (verb) sin (det) ? (punct) " (punct) ol
(verb) Ailill (Prop) .
" (punct) Aithgén (verb) " (punct) , (punct) ol (verb) sí (pronoun) .
Fosceird (verb) Ailill (Prop) issin (prep) n - abaind (noun) sís (adverb) .
Ro (verb_part) airigestar (verb) Fráech (Prop) anísin (pronoun) .
```

Figure 7: Part-of-speech tagged corpus of *TBF*

3 Possible Further Developments

As this project was developed for experimental purposes, the machine-readable lexicon and pos-tagger are merely prototypes and therefore fairly rudimentary. There is much scope for further development however, which, given the promising results at this early stage of experiments, could be a worthwhile basis of future research. The resources developed may be enriched in multiple ways:

Line numbers: in the lexicon, I have omitted the original line numbers that Meid included in his vocabulary. This information would be useful if it were reintroduced as it would mean the user would not have to revert back to the original text to ascertain what part of the text the analysis results are taken from.

Multiple entries: as the pos-tagger currently does not have a disambiguation function, there are no multiple entries in the lexicon. For example, *baí* is a form of the verb *atá* 'that is' and it is also the plural form of the noun *bó* 'cows'. In these cases, I included only the more frequently used form from the text in the lexicon and removed the other entries. This functionality, of course, would need to be implemented for more extensive work on the text.

Multiword recognition: The entries in the lexicon are recorded on a one-word-per-entry basis. In other words, multiword terms such as *Bé Find* 'Fair Woman' are not recognised as one unit of meaning. More work is required on the pos-tagger before it is intelligent enough to deal with such multiword forms.

Morphological analyser: The development of a morphological analyser would greatly benefit textual analysis of TBF. While the current lexicon holds some information about spelling variants and inflected forms, there are still many inflected forms that are not accounted for. For example *mbó* is the form of the word *bó* 'cow' when nasalised, for instance after an article in the genitive plural. Many words are mutated through nasalisation or lenition for grammatical purposes. Due to the highly inflectional nature of Medieval Irish, it would be inefficient to record all possible inflected verb forms, nouns, adjectives etc. in the lexicon. Morphological analysers are developed upon this type of linguistic information to allow for pulling apart of inflected forms, recognising inflections, and mutations such as lenition, nasalisation, as well as suffixation, thus identifying the root form of the word. Ideally, we would only need to store the root form of each word in the lexicon and thus ensure it does not increase too much in size.

An important factor to consider during further developments for this project is the effort to limit the number of lexical entries in the lexicon. The main reason for this is to ensure computational efficiency during search and match processing. A smaller lexicon is easily managed and maintained within high quality levels. As described above, a morphological analyser can assist in achieving this goal by identifying and linking inflected forms to base form entries. The sophistication of the lexicon can be sustained with the inclusion of rich meta-data within each entry.

4 Requirements for Similar Continued Research

Text resources: Correct format of text resources for this kind of computational processing is important. Scanned PDF files are not sufficient for these purposes as they are just images of text. The text must be in machine-readable format. If the text is only available in a scanned version, it is possible to extract the text

using Optical Character Recognition software (OCR) processing. If the language of the text being processed is not an available option in the software, other languages which support similar diacritics can often be useful. For example the Czech option works well with Medieval Irish characters. Text files should be saved with UTF-8 encoding to support the correct display of accented characters.

Technical Skills: Ideally, the user of these tools would be a computational linguist with sufficient linguistic skills in the language being processed. However, in the absence of that combination, users with technical skills—preferably computer-programming experience—could apply their own linguistic knowledge of the language or collaborate with a trained linguist. The *NLTK* website provides extensive documentation and an online book which guides users through NLP techniques in a practical manner. Many of the early chapters of the book do not require previous programming experience and there are exercises at the end of each chapter for users to test their newly acquired skills and knowledge.

*Collaboration***:** Projects such as this are most successful when based on collaborative research across different disciplines. As the *NLTK* book points out, NLP is of interest to academics from humanities computing and corpus linguistics through to computer science and artificial intelligence. Digital Humanities is a more recent discipline which can also bridge the gap between humanities and computer science. In Ireland, the efforts of the DHO—Digital Humanities Observatory (Royal Irish Academy 2008–2011), *CELT—Corpus of Electronic Texts* and *eDIL—Electronic Dictionary of the Irish Language* (Toner & al. 2007) are notable in this regard.

5 Findings

The results of this project have proved interesting in many ways. The overall aim was to assess how conducive Medieval Irish is to computational linguistic techniques. Through the application of tools made available through *NLTK*, along with Python coding techniques demonstrated by the toolkit, it is clear that Medieval Irish works well with today's advanced technology.

We can see from the various experiments presented that it is possible to gain benefit from these tools for textual research. In this context, it is important to note that the areas of study which can potentially benefit extend well beyond the analysis of one medieval text. These tools can assist with work on many types of text-based research.

We can also see how these tools can reduce analysis time in such research. Many of the tasks demonstrated in this project would take weeks, if not months to perform manually. Depending on the technical proficiency of the computational linguist performing these tasks with *NLTK* or *Python*, some of these tasks can take just a few minutes to complete. Others may take a couple of hours. Overall, the savings in time spent cannot be overstated.

The attraction of reducing analysis time also encourages attempts at some tasks that may otherwise appear too big to tackle. Undertaking such seemingly mammoth tasks also poses a risk when there is no guarantee of interesting results. The application of computational techniques will remove the risk of time spent in vain and possibly encourage more extensive research projects.

It is clear that the possibilities for exciting research are more extensive when combining skills from various disciplines. This project has highlighted just a few benefits that researchers in humanities can gain from applying computational techniques to written natural language.

References

Bird, Steven, Edward Loper & Ewan Klein 2009a *Natural Language Processing with Python*, Sebastopol, CA: O'Reilly Media Inc.

CELT = Färber, Beatrix 2001 *CELT Corpus of Electronic Texts*, University College Cork <*http://www.ucc.ie/celt/published/G301006/*> [accessed 2 September 2011].

Jurafsky, Daniel & James H. Martin [2]2009 *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, Upper Saddle River, NJ: Pearson Prentice Hall.

Meid, Wolfgang, editor 1974 *Táin Bó Fraích*, Dublin Institute for Advanced Studies.

Meid, Wolfgang, editor [2]2009 *Die Romanze von Froech und Findabair Táin Bó Froích*, Innsbrucker Beiträge zur Kulturwissenschaft.

NLTK = Bird, Steven, Edward Loper & Ewan Klein 2009b *Natural Language Toolkit* <*http://www.nltk.org*> [accessed 2 September 2011].

Python Software Foundation <*http://www.python.org*> [accessed 28 September 2011].

Royal Irish Academy 2008–2011 *Digital Humanities Observatory* <*http://dho.ie*> [accessed 2 September 2011].

Toner, Gregory, Maxim Fomin, Thomas Torma & Grigory Bondarenko 2007 *Electronic Dictionary of the Irish Language* <*http://www.dil.ie*> [accessed 2 September 2011].