Language and Technology in Wales: Volume I

Editor: Delyth Prys

PRIFYSGOL BANGOR UNIVERSITY

Ariennir gan **Lywodraeth Cymru**
Funded by **Welsh Government**

This ebook was first published in 2021 by

Bangor University, College Road, Bangor, Gwynedd LL57 2DG

www.bangor.ac.uk/

International Book Number (ebook):

ISBN 978-1-84220-188-6.

Design and proofreading assistance from Stefano Ghazzali and Dr Gareth Watkins.This book is also available in Welsh under the title *Iaith a Thechnoleg yng Nghymru: Cyfrol 1*, number ISBN 978-1-84220-189-3.

Language and Technology in Wales: Volume 1

Editor:

Delyth Prys

Contributors :

Kepa Sarasola UNIVERSITY OF THE BASQUE COUNTRY

Iñaki Alegria UNIVERSITY OF THE BASQUE COUNTRY

Olatz Perez-de-Viñaspre UNIVERSITY OF THE BASQUE COUNTRY

Geraint Palmer CARDIFF UNIVERSITY

Padraig Corcoran CARDIFF UNIVERSITY

Laura Arman CARDIFF UNIVERSITY

Dawn Knight CARDIFF UNIVERSITY

Irena Spasic CARDIFF UNIVERSITY

Dewi Bryn Jones BANGOR UNIVERSITY

Sarah Cooper BANGOR UNIVERSITY

Myfyr Prys CYMEN CYF

Vigneshwaran Muralidaran CARDIFF UNIVERSITY

Keeziah O'Hare CARDIFF UNIVERSITY

Gruffudd Prys BANGOR UNIVERSITY

Gareth Watkins BANGOR UNIVERSITY

Jonathan C Roberts BANGOR UNIVERSITY

Peter W. S Butcher BANGOR UNIVERSITY

Robert Lew ADAM MICKIEWICZ UNIVERSITY

Geraint Rees UNIVERSITY OF SURREY

Nirwan Sharma THE OPEN UNIVERSITY

Ana Frankenberg-Garcia UNIVERSITY OF SURREY

Leena Sarah Farhat BANGOR UNIVERSITY

William John Teahan BANGOR UNIVERSITY

Contents

Preface

JEREMY MILES MS, MINISTER FOR EDUCATION AND WELSH LANGUAGE

When I was appointed Minister for Education and the Welsh Language, I noted increasing the use of our language as a key focus of my work. Technology has a central role to play in that – particularly in light of COVID-19, which has recently changed so many of our lives.

Looking at the Welsh Government's *Welsh Language Technology Action Plan*, you will see that our emphasis is on Welsh language speech, translation and artificial intelligence technologies. We have made sure that as much of the work as possible is freely available for use and reuse, under an open licence. I very much hope that this approach will make it attractive and easy for companies to adopt the components created in their products and to offer those products proactively in Welsh to customers, clients and citizens alike. Making Welsh easier – that is the trick in increasing Welsh language use. And making Welsh easy is what we are doing at the Welsh Government in incorporating Good bilingual User Experience into our technology procurement systems.

The detailed research and other work you will read about in the volume before you has created resources of which we can all be proud - resources of which our colleagues in other linguistic communities can also avail themselves.

Technology is constantly evolving, and Welsh must evolve with it. I look forward to seeing further developments in the field and thank all the developers and all others involved.

Introduction

DELYTH PRYS, BANGOR UNIVERSITY

This volume is based on some of the main papers presented during the Language and Technology in Wales 2020 Academic Symposium. The symposium was held at the request and with the support of the Welsh Government, with the intention of gathering some of the main researchers in Wales, sharing knowledge and promoting further research. Due to the Covid restrictions, the Symposium had to be run in virtual form, which itself says a lot about the importance of technology in the modern world, and its ability to facilitate communication with each other.

Language Technologies include speech technologies (used for example to transcribe a lecture or a conversation from speech to text, or create synthetic voices), text technologies (which include grammar checkers and recognizing emotion or meaning in texts), natural language processing (for example analysis of syntax and morphology, terminology extraction and text anonymization). Deep Learning neural methods are now used in all these technologies and are getting much better results than methods previously in vogue. This on the other hand poses an additional problem for less resourced languages such as Welsh. To use the new neural methodologies, huge amounts of various types of data, in the form of text or speech, are needed to identify the necessary patterns and train language models. This data is usually lacking in less resourced languages. These languages include some that we have traditionally called 'minority' languages; languages with few speakers; and languages of disadvantaged communities, such as those of India and Africa. The languages of these deprived communities may have millions of speakers, but they may also have low literacy, or lack a reliable electricity supply and internet access - some of the essentials of modern communication.

Having communication tools based on language technologies is now seen as key to the survival and prosperity of human languages. Without computer programs that can deal with a certain language, it is not possible to use the full range of that language in digital contexts and on the world-wide-web, and speakers must turn to one of the major languages to give verbal commands to their electronic gadgets, to access text to speech services, or to get help with writing correctly. The danger of extinction for less-resourced languages is therefore a real one, and the topic of language revitalization through language technologies has become an important one for the language and policy planners of governments concerned with protecting the languages and civil rights of their communities. That's why researchers in Wales are trying to tackle these challenges, to offer a solution for Welsh, and other languages in similar situations.

One interesting research question in this area is whether language technologies can be developed for these languages in the same way as is done for major world languages? An important part of this is discovering how to locate sufficient data in less resourced languages to meet the requirements of Artificial Intelligence and Deep Learning. Volunteers in Wales have worked miracles in projects such as Mozilla's Common Voice, recording their voices so that enough speech data exists. The willingness of public bodies to share texts such as their translation memories is also part of the solution. Other possible solutions are to use machine translation methods to create 'synthetic data' where there is little real data available in a language, especially for some subject areas, and to use transfer learning methods between a language where some technology has already been developed and another language where the technology is not yet available. We are currently in the middle of an intense period of innovation and progress, and it's a very exciting time to be a researcher in this field.

In Wales researchers at Bangor University have been conducting language technology research for several years, and more recently there has been similar activity at Cardiff University and the

University of South Wales. We also have an active community of volunteers, who create and contribute Welsh data, and a base of local SMEs, such as Cymen translation company in Caernarfon, that incorporate research results into their products, thereby promoting the economy and social well-being. There is still much to do to improve the teaching of language technologies in Wales, and the new MSc Language Technologies course at Bangor University is a step in the right direction. Much remains to be done, starting with increasing the emphasis on teaching coding to children in our primary and secondary schools, and bringing the linguistic and computing elements together at undergraduate and postgraduate level in our universities.

Language technologies research is essentially an interdisciplinary activity, and in order to strengthen collaboration between various researchers and other stakeholders in Wales, a National Language Technologies Network was established in 2020 (https://rhwydwaith.techiaith.cymru/). All are welcome to join this network, to meet and exchange ideas, not only academics from different departments and universities, but also industry stakeholders and policy makers, not forgetting the many volunteers, a community which is essential to enable a small language such as Welsh to develop advanced language technology resources and tools for itself. Creating language technologies is also a global activity, of interest to large multinational corporations such as Google and Microsoft, as well as governments, stateless communities and language activists. Following Jill Evans MEP's successful European Parliament proposal in September 2018 on linguistic equality in the digital age (https://www.europarl.europa.eu/doceo/document/TA-8-2018-0332_EN.html), the European Commission set up a project to produce an agenda and roadmap to achieve full digital equality for all European languages by 2030 (https://european-language-equality.eu/). This means a new emphasis on developing appropriate language technologies over the coming years, and Welsh researchers in universities and industry need to be prepared.

This volume is a contribution to the development of the field in Wales. It is published bilingually under a permissive open licence which will mean that the parallel text of the volume in Welsh and English will be added to Bangor's permissively licenced corpus and contribute to Big Data which is so important to the development of language technologies. The volume opens with a chapter on the work to develop language technologies for Basque, another small language in a very similar situation to Welsh, by Kepa Sarasola, Iñaki Alegria and Olatz Perez-de-Viñaspre from the University of the Basque Country. This chapter provides a broader context to our efforts in Wales and reminds us of the value of working together and sharing ideas across geographical boundaries. Researchers from several departments at Cardiff University, namely Geraint Palmer, Padraig Corcoran, Laura Arman, Dawn Knight, Irena Spasic, Vigneshwaran Muralidaran, and Keeziah O'Hare, who have been researching Welsh word embeddings and an implementation of Porter's English stemmer for the Welsh language contribute other chapters in the volume. Jonathan C Roberts and Peter WS Butcher of Bangor University, in collaboration with Robert Lew of Adam Mickiewicz University, Nirwan Sharma of the Open University and Geraint Rees and Ana Frankenberg-Garcia, both from the University of Surrey, UK, discuss adapting programs originally developed for English, and specifically the Collocaid program. Bill Teahan and Leena Farhat of Bangor University are pursuing a similar approach when discussing adapting the TAWA toolkit for the Welsh language. Also from Bangor University are chapters from Dewi Jones and Sarah Cooper discussing bilingual text-to-speech, and Gruffudd Prys and Gareth Watkins discussing a new library for processing Welsh. A chapter by Myfyr Prys, a former KTP and Smart researcher between Bangor University and Cymen, now working for Cymen, discusses the development of English<>Welsh machine translation for a translation company.

This is a snapshot of some of the research topics currently being pursued by language technologists in Wales and Europe. It is hoped that this volume will be useful to further research in the field, educate students, and to contribute to the development of language technology tools for the benefit of Welsh and other less resourced languages.

Language Technology for Language Communities: An Overview based on Our Experience (2020)

KEPA SARASOLA

IXA group, University of the Basque Country (UPV/EHU)

IÑAKI ALEGRIA

IXA group, University of the Basque Country (UPV/EHU)

OLATZ PEREZ-DE-VIÑASPRE

IXA group, University of the Basque Country (UPV/EHU)

IXA is a research group that has been working on language technology, mainly on Basque, over the last 32 years. As a result of years of collaboration with the Basque and other language communities we conclude that language technology is an important factor for language development. Some initial core work is needed: 1) standardization and 2) generation of open contents. Bearing these requisites in mind, we propose the definition of a BLARK (Basic Language Resource Kit) to identify a minimal set of basic resources, and then we suggest tools for their adaptation to different languages depending on the size of their speakers' community and digital resources. Finally we present some successful experiments that allow us to be optimistic about the potential use of neural networks, deep learning and BERT linguistic models for less resourced languages.

1 INTRODUCTION

The IXA group (www.ixa.eus) is a research group created in 1988 with the aim of laying foundations for research and development of Natural Language Text-Processing (NLP) and Human Language Technology (HLT) for the Basque language. It is now a large multidisciplinary group composed of computer scientists and linguists.

Two distinguishing features of IXA are that it deals with a less-resourced language (Basque) and that it combines classic linguistic modeling and data analysis with innovative probabilistic and machine-learning approaches to NLP.

At the very beginning, thirty nine years ago, our first funding was awarded for the creation of a translation system for Spanish-Basque. But after some preliminary studies we realized that it was more important to concentrate our efforts in creating basic tools and resources for Basque (morphological analyser/generator, electronic dictionaries, annotated corpora, semantic databases etc.) that could be used later on to build many other general language applications, rather than creating an *ad hoc* and extremely complicated MT system. This realization underlaid our strategy to make progress in the adaptation of language technology for Basque.

Since then, our research has resulted in state-of-the-art technology for robust, broad-coverage natural-language processing for Basque. These technologies/resources include a spell-checker (Xuxen), Basque Wordnet (BasqueWN), the corpus of Science and Technology (ZT corpus), a syntactically-annotated corpus (EPEC), a Spanish-Basque MT system (Matxin), a NLP pipeline for text processing (IXA-pipes) and an opinion-mining tool (Behagunea).

Based on our experience with NLP for less-resourced languages [1], we have been collaborating for many years with two kinds of language communities:

- Those working on promoting the Basque language (dictionaries, language learning methods, Wikipedia, keyboards and interpretation tools for smart phones)
- Those working on less-resourced languages (such as Quichua, Nahuatl, Spanish in Cuba) in order to help language communities with the technological development of those language.

Borin [2] pointed to the potential HLT had to offer lesser-known languages and describes the linguistic diversity in the information society. He cites Ostler: "*a language will not get by in the world of today unless it is equipped with a parser and a multi-million-word corpus of text*". He analyzed the relation between the sociology of language and HLT, and gave us some strategic considerations.

In our opinion while technology may be an important factor for language development, there is also some core work which must be implemented in advance (or concurrently):

- Standardization: the fragmentation of the community into dialects makes it difficult to generate written text. Standardization must have priority if the goal is to effectively promote the use and status of the language. Dialects, of course, have their role, especially in oral domains and informal registers.
- Digital content: without a minimum basis (consisting mainly of scholarly books, translations and Wikipedia) it will be impossible to generate useful tools for the language community.
- Open content and open-source software: the decision to promote the production of open content and the use of open-source tools is crucial to ensure an incremental and sustainable development of this technology.

Kornai [3] argues that the danger of digital language death is underestimated and concludes that less than 5% of all languages can still ascend to the digital realm. He introduces a four-way classification for the languages: digitally thriving, vital, heritage and still languages.

Bearing these requisites in mind we present the concept of BLARK and its adaptation to the size of the community. A BLARK for a language [4] is the minimal set of basic resources (software modules, corpora, dictionaries, etc.) that are necessary to do further research and development in the field of language technology.

The chapter is structured as follows. After discussing the relevance of several elements such as the role of a language community, the level of standardization and the amount of text available (Section 2), we present related work (Section 3). In Section 4 we present the key tasks, resources and applications to be implemented in a concrete roadmap for low-resourced languages, including corpus compilation, digital dictionary, spell-checker, morphological analyzer, corpus annotation, POS tagger and text-mining. We also present a project of machine translation to make the use of health records in Basque easier. Finally, in Section 5, we present our conclusions.

2 RELEVANCE OF COMMUNITY, STANDARDIZATION AND DIGITAL CONTENT

Standardization of the language is a prerequisite for successful use of the written language.[2]

In Basque there are approximately 800,000 speakers and six dialects. These dialects are very different from each other. In 1968 the Academy of the Basque Language decided to create Standard Basque. Eventually, following some years of discussion, the standard language (named '*Batua*') was widely accepted and it is now the standard and the language model used in (almost) all formal texts: school, university, administration, the internet etc. TV and radio journalists and academics speak in a standard way.

As Hualde and Zuazo [5] say "By any criterion that we may choose, the standardization of Basque in recent years has been a very successful project. Nowadays, standard Basque, which was not developed until the late 1960s, is used in education at all levels, from elementary school to the university, on television and radio, and in the vast

[2]It may be argued that it is not a necessity for speech processing but most of the speech-to-text systems need resources based on standard texts.

majority of all written production in Basque. This success in the societal acceptance of standard Basque is most remarkable given the fact that there is no administration common to all territories where Basque is spoken (divided as they are between Spain and France and even, within Spain, into two separate administrative regions with different legislation regarding the Basque language) and that Basque speakers are almost always fully bilingual in either Spanish or French, so that the existence of a standard Basque language is not strictly required for communication beyond the local level."

We want to underline the relevance of the work done by the linguistic community in this process; it was the community who pressed for an academic/political decision to accept the standard, and it was the community who generated new resources using the standard (books, magazines, dictionaries, a newspaper, Wikipedia etc.). The work has been especially important in the role that the Basque schools (*Ikastolak*) had in the recovery and standardization of Basque [6].

It was very important for us that the Basque standard had been defined and widely accepted before our research group started to develop new NLP tools or applications. When we needed linguistic knowledge we did not need to create it for ourselves, since this work had been done previously. We had no need to deal with different dialectical variants for a word, no need to choose one of those variants, as the Academy of the Basque Language (Euskaltzaindia) had already done it. Now when we need corpora for learning or for inference, finding adequate text is easier. Consequently, this aspect has become a key factor for success because a text corpus is the raw material for the current main technological paradigm: data-driven language engineering.

Later on, when we were collaborating with academics or other communities in order to develop technology for low-resourced languages we were more highly aware of the importance of the standardization of a language. For example, unifying efforts for Quechua is a difficult task because in Peru, Bolivia and Ecuador they use different variants of the same language.[3]

As we will explain below, Wikipedia is becoming a key resource, not just as a single text corpus but even as a suitable basis for the development of new tools and applications. Unfortunately, sometimes there is no agreement among local communities to define and promote a standard variety of the language. The consequence is usually a smaller Wikipedia, inefficient use of human resources, and a more divided community, e.g. the decision of whether to use classical Nahuatl or not is still under discussion.[4]

Dialects and variants are also an important matter for the language community,[5] but in our opinion standard language is a priority for text processing.

3 BLARK AND OPEN SOURCE

Krauwer [4] proposed a "Basic Language Resource Kit (BLARK)" as a roadmap of tools to be developed for each language using the terminology defined in a joint initiative between ELSNET (European Network of Excellence in Language and Speech) and ELRA (European Language Resources Association) in 1998. In all these works a list of basic resources and tools are listed. The term BLARK has been widely adopted and it is used in a large number of

[3]We know that the variants can be considered different languages, and therefore communities have to decide if they prefer joint efforts or to work separately.

[4]https://meta.wikimedia.org/wiki/Proposals_for_closing_projects/Closure_of_Classical_Nahuatl_Wikipedia

[5]Social networks (especially Twitter) are becoming an important resource for identification/treatment of variants/dialects

papers in the area. Recently Mager et al. [7] describe the challenges of language technologies for the indigenous languages of the Americas and Hassani [8] proposes the design of a BLARK toolkit for multi-dialect Kurdish.

Streiter et al. [9] reported on HLT projects for non-central languages and proposed instructions for funding bodies and strategies for developers. They used the term "non-central" and underlined the importance of making use of free software to improve the results. Their chapter about benefits and unsolved problems when using open-source software for non-central languages is very interesting. Forcada [10] pointed out the opportunity to use open-source machine translation for minor languages.

The ELSNET network of excellence prepared definitions for a "language-resources and evaluation roadmap". HLT products are classified into three subsets: Language Resources, Language Processing and Language Usage. (In our proposal, these appear as Language Resources, Language Tools and Language Applications).

Based on several indicators we have proposed six levels to classify the adaptation of the languages to the technology [1]:

1. English: Around 45% of all web pages are written in English. Almost all the HLT applications are available for English. Most research is carried out through testing on English texts.
2. The next top 10 languages that cover almost 50% of Internet users. These are the languages for which active resource development continues and most major companies on the internet support them. Streiter et al. (2006) call them the central languages.
3. Around 70 languages with any HLT resources registered. Sometimes they are named non-central languages.
4. Around 300 languages with any lexical resource on-line registered in yourdictionary.com. It is almost the same set of the languages that are in Wikipedia or the set of languages that have defined their standard. The term low-resourced (or lesser-resourced) language is used to be applied to these languages (and to the previous level also).
5. About 2,000 languages, namely 2,014 languages that have writing systems [2].
6. The big bag including languages in the world that are only spoken (more than 4,000). Most of them can be considered endangered languages.

In the next section we try to define, according to our experience, the most important resources, tools and applications to be developed as a roadmap for level 4 and 5 languages. These languages can fit with those classified as Vital by [3].

In addition to this we want to stress how the linguistic and academic communities can cooperate in their development. In some cases, i.e. natural disasters, it can be important to give a quick response [11], but in general it is better to set a plan depending on the situation of the language: number of speakers, their connectivity, digital resources, integration in education etc.

If there is an important group of internet users, developing collaborative tools is a very productive path. Tools on Wikimedia (Wiktionary, Wikipedia etc.) are the best known, but there are other tools, such as the crowdsourcing platforms [12] which can be used by language communities. When internet users are scarce, finding collaboration from academia and schools is more suitable.

4 KEY RESOURCES AND APPLICATIONS

In the next subsections we propose a concrete roadmap for low-resourced languages, starting with the most basic resources/tools/applications. We have selected mainly open-source resources and tools. This roadmap is based on our experience and the proposal by Streiter et al.[9].

We will not include machine translation among the applications because it needs more resources than are available to level 4 and 5 languages. Anyway, if there are closely related languages with more resources, a machine translator among similar languages can be built without great effort. Apertium[6] [10] is a good example in this area.

It is important to underline that the tools we propose are not based on machine learning or deep learning because these approaches need more data and computational resources. However, during the last few years important advances have been made in applying neural techniques to languages that do not have enough data.

We think that for a robust development of language technology the next steps come before neural or statistical experiments.

5 CORPUS COMPILATION AND DIGITAL DICTIONARY

5.1 Corpus.

A monolingual corpus is the first basic resource for language technology. Its most important feature is its size, but there are other features to be taken into account: normalization/variants, domain, single/multiple sources etc. It can be a big project if the task is to build a "national corpus" or a "monitor corpus" including metadata (XML/TEI is the standard way for this) and additional tools.

Wikipedia is a good option for corpus extraction, but in the cases where Wikipedia does not exist for a language or it is too short, dealing with *web as a corpus* techniques may be a good option if substantial texts are available on the internet. Where such texts are not available, scanning texts or collaboration with editors and teachers/academics are the remaining options.

Web as a corpus techniques were described by Kilgarriff and Grefenstette [13], and Webcorp[7] is an interesting tool for this aim. Sometimes the program needs to be adapted to suit the particular linguistic features of a language [14].

When scanning of documents or compilation of digital files is necessary, it is important to preview and measure the real scope of the work, which may include: compiling documents or files in different formats, dealing with licenses and legal issues, scans or format conversion, OCR, insertion of metadata etc. From our experience [15] this is a major task, much bigger than was previously expected. Gutierrez-Vasques et al. [16] offer an example of a bilingual compilation.

There are also global projects for building a corpus for multiple languages [17],[18] .

[6] https://www.apertium.org

[7] http://www.webcorp.org.uk/live/

Based on the corpus, first/initial applications can be developed, for instance, examples for language learning, dictionary of frequencies, basic games (looking for short words, long words, palindromes...). Natural Language Toolkit (NLTK)[8] is a very interesting tool set for the development of such applications.

5.2 Digital Dictionary

A dictionary is a key tool for students. This is a very important tool. From our experience, together with the spell-checker, it is the most practical application that we have developed. When available, Wiktionary can be the basis, but it can also be built from a corpus or from a previous dictionary.[9]

A corpus may be helpful for quality testing and to find new entries, but the best option is a previous lexicographic work. From our experience we know that in some communities a digital dictionary exists, but it is not available on the internet or it has a proprietary license. The conversion of such dictionaries into a multimedia online dictionary based in a lexical database is a very important task.

A good experience for us was the semiautomatic transformation of the Cuban "Diccionario Básico Escolar"[19].[10] For Basque, Euskalbar[11] (an add-on for browsers which sends concurrent queries to existing online dictionaries and corpora, and shows all the results simultaneously) is a key application for the community.

Based on the dictionary, new applications can be developed, especially for students. In that way, our group was involved in building the Basque version of *Apalabrados*.[12]

6 SPELLING AND MORPHOLOGY

As mentioned, a spell checker is one of the most useful applications for a language. Students, teachers, journalists, writers, etc. all use it. It is even more necessary when the written system for a language is in development. Furthermore, in the case of Basque it has been a very effective tool in the standardization process.

A spell checker may be generated from a large (good) corpus, but its quality and coherence would be better if its construction were based on a morphological analyzer. This latter approach is mandatory for morphologically rich languages.

A morphological analyzer obtains, for each word, its possible morphological segmentations, some lemmas that use it, and the part-of-speech category associated with each word-form. Based on this, the speller decides whether words without morphological analysis are mistakes or variants.

To build the analyzer it is necessary to specify: (1) the set of lemmas with their categories, (2) the affixes, (3) the morphotactics describing valid linkings among lemmas and affixes and (4) the morphophonological changes produced when linking lemmas and affixes. The first specification, the set of lemmas, may be obtained from the digital dictionary and the others from academics or from a formal basic grammar. Tools have been developed to put

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

[9] yourdictionary.com presents links to on-line lexical resources (http://www.yourdictionary.com/ languages.html for 307 languages.

[10] http://ixa2.si.ehu.es/dbe

[11] https://addons.mozilla.org/eu/firefox/addon/euskalbar/

[12] http://www.apalabrados.com/

these all together. We used the two most popular tools, namely *foma*[13] and *hunspell*.[14] The first one [20] is linguistically better motivated and simpler in respect of creating a description of a language's morphology, but using the second one has been more successful because the description can be directly integrated as a spell-checker in many software packages (Libreoffice, Mozilla…).[15] For Basque [21] and for Quichua [22] both options have been combined, by creating the first description using *foma* and then automatically converting it to *hunspell*.[16]

Of course, the community has an important role to play in the construction and distribution of the spell-checker in respect of testing the tool, its dissemination, helping new users install it on their computers, sending feedback on errors or missing lemmas etc.

7 ANNOTATION, POS TAGGING AND TEXT-MINING

Raw text corpora are a nice resource to develop very basic NLP applications, but corpora annotated with morphological, syntactic or word meaning information opens the door to (semi-) automatically building part-of-speech taggers, and tools for text mining.

For instance, we built EPEC (Reference Corpus for the Processing of Basque) for Basque,[17] which is a 300,000-word corpus of standard written Basque. It was manually tagged at different levels: morphosyntax, syntactic phrases, etc. It has already been used for the construction of some tools such as a POS tagger.

The POS tagger is another key tool, along with the digital dictionary and the spelling checker, because it is a mandatory preparatory step for text mining: fact extraction, identification of entities (persons, places, organizations), extraction of terminology, text simplification, etc. The tagger assigns to each word in a text its part-of-speech (POS), based on its definition (or morphological analysis) and its context.

As we have said before, it is possible to build many applications for text mining based on POS tagging, and more powerful tools too. The IXA pipes framework [23] is an example of how to build these new tools easily. It is a modular set of Natural Language Processing tools (or pipes) which provide easy access to NLP technology for several languages. It offers robust and efficient linguistic annotation, which is very useful in text-mining. This open technology is easily adaptable to any other language, the only prerequisite is access to a linguistically annotated corpus.

8 BASQUE, NLP AND CLINICAL DOMAIN

The use of machine translation tools between languages in today's society is common and widespread. In 2019 the Ixa group and Osakidetza (The official Organization for Health in the Basque Country) saw the opportunity to develop a tool adapted to the clinical field by using the new technological conditions (use of the successful paradigm of neural networks in machine translation) and leveraging the new professional conditions (increase of

[13]https://fomafst.github.io/

[14]http://hunspell.github.io/

[15]https://addons.mozilla.org/en-US/firefox/language-tools/ List of the spelling-checkers supported by Mozilla.

[16]Another matter is that Microsoft Office is the main tool for any users. Streiter et al. (2006) discuss this.

[17]Our steps on standardization of resources led us to adopt TEI and XML standards as a basis for linguistic annotation (Artola et al., 2009).

bilingual staff who want to work in Basque and significant number of new young doctors trained in Basque at the university).

We had been working since 2015 on translating SNOMED CT Terminology into Basque [24], [25] and on Neural Machine Translation of clinical records from Basque and into Basque [26], [27].

Neither translation nor even the development of automatic translators are the final objective in Basque Country official plans, but they are potentially useful tools to get to it. The objective of the Basque Country official plans, as well as that of Osakidetza, is to increase the presence and use of the Basque language in its everyday clinical histories, and it must be demonstrated whether this tool will contribute to this goal. In fact, Itzulbide has been launched as a research project based on the hypothesis that if the general domain MT system is taught to translate in the clinical field, in the future we will have a fast and reliable translation tool. Within a few years it will be seen whether this hypothesis is true.

The Itzulbide project began in June 2019 and the promoters of this project (Ixa Group of the UPV/EHU and the Osakidetza Itzulbide working group) have begun to carry out the open presentations of the centre-to-centre project to clarify the opinions and doubts of the professionals and to collect the contributions of the professionals. At the time of writing of this text, 68 professionals from different specialties and categories collaborated in the project, creating bilingual clinical texts. Encouragement and thanks to all the participants!

The "Itzulbide" Automatic Translator project does not prevent or condition the other complementary specific language objectives, nor the normalization measures currently included in the Osakidetza's Basque Language Plan.

If its usefulness is demonstrated, the tool will be integrated into the information system of Osakidetza, but in addition, the development of this tool could extend to the entire healthcare community (professionals of public and private companies, pharmacists, university students and professors, and non-university health residents, professional associations) and to the geographical scope of the Basque language. It can also be a help tool for professionals who are learning Basque. In summary, the possible use of Itzulbide could go beyond clinical history.

A project of this type can generate doubts, but we will test and measure whether this tool brings us closer to the objectives of the Basque Country's Language Plan.

9 GOOD NEWS ON MACHINE LEARNING, DEEP LEARNING AND BERT LANGUAGE MODELS

A priori, machine learning, deep learning and BERT language models are not useful for low resourced languages because these approaches require more data and computational resources. However, as mentioned above, during the last few years important advances have been made in applying neural techniques to languages that do not have enough data.

There have been significant advances, even for less-resourced languages, in several areas: lexicon extraction [28], morphology induction [29], POS tagging [30], machine translation [31]. In most of the cases cross-lingual learning is used, but good results are also obtained even where only monolingual corpora are used, which is good for languages with few parallel resources. However Artetxe et al. [32] argue that a scenario without any parallel data and abundant monolingual data is unrealistic in practice.

In that way, pre-trained Basque monolingual and multilingual language BERT models have recently proven to be very useful in NLP tasks for Basque, even though they have been created with a corpus that is 500 times smaller than the English one and with a Wikipedia that is 80 times smaller.[18] Word embeddings and pre-trained language

[18]https://www.ehu.eus/ehusfera/ixa/2020/09/30/ixambert-good-news-for-languages-with-few-resources/

models allow us to build rich representations of text and have enabled improvements across most NLP tasks. Unfortunately, they are very expensive to train, and many small companies and research groups tend to use models that have been pre-trained and made available by third parties, rather than building their own. This is suboptimal as, for many languages, the models have been trained on smaller (or lower quality) corpora. In addition, monolingual pre-trained models for non-English languages are not always available. At best, models for those languages are included in multilingual versions, where each language shares the quota of substrings and parameters with the rest of the languages. This was particularly true for smaller languages such as Basque, but last April our monolingual models for Basque produced much better results than publicly available versions in downstream NLP tasks, including topic classification, sentiment classification, PoS tagging and NER [33]. The original BERT language model for English was trained in 2018 using the Google books corpus,[19] which contains 155 billion words in American English, and 34 billion words in British English. The English corpus, therefore, is almost 500 times bigger than the Basque one. The composition of the Basque Media Corpus (BMC) used in that experiment was as follows:

Table 1: The composition of the Basque Media Corpus (BMC)

Source	Text type	Million tokens
Basque Wikipedia	Encyclopaedia	35M
Berria newspaper	News	81M
EiTB Television	News	28M
Argia magazine	News	16M
Local news sites	News	224.6M

Moreover, later on, in September 2020, IXAmBERT, a multilingual language model pretrained only for English, Spanish and Basque that re-uses the same corpus of the monolingual Basque model and adds the English and Spanish Wikipedia with 2.5G and 650M tokens respectively (80 and 20 times bigger than the Basque Wikipedia) has successfully been used in a Basque Conversational Question Answering system. This transfer experiment could be already performed with Google's official mBERT model, but as it covers that many languages, Basque is not very well represented. The good news is that this model has been successfully used to transfer knowledge from English to Basque in a conversational Question/Answering system [34]. These works set a new state-of-the-art in those tasks for Basque, and all benchmarks and models used in this work are publicly available.[20]

10 CONCLUSIONS

Language technology is a powerful aid for low-resourced language communities, in order to revitalize the language and to effectively promote its use.

But there are some prerequisites to allow language technology to be used. A language community needs to be ready to distribute and disseminate the LT tools. The existence of a standard for the language and a wide acceptance of it will definitively aid the development of new NLP tools and their effectiveness.

[19]https://www.english-corpora.org/googlebooks/

[20]https://huggingface.co/ixa-ehu/ixambert-base-cased

Completing tasks such as corpus compilation and corpus annotation and creating tools such as a digital dictionary, spell-checker, morphological analyzer, POS tagger and text-mining tools are the first steps to be faced. We have presented our fruitful experience dealing with Basque, and put forward some suggestions for other languages that want to design a roadmap for language technology.

Acknowledgements

The authors of this work have received financial support from these two projects: DOMINO (MINECO-Spain, PGC2018-102041-B-I00 (MCIU/AEI/FEDER, UE) and TANDO (Basque Country, ELKARTEK20/49) .

References

[1] Iñaki Alegria, Xabier Artola, Arantza Diaz De Ilarraza and Kepa Sarasola. 2011. Strategies to develop Language Technologies for Less-Resourced Languages based on the case of Basque. In Proceedings of 5th Language & Technology Conference: HLT as a Challenge for Computer Science and Linguistics. LTC, Poznan, Poland, 42-46.

[2] Lars Borin. 2009. Linguistic diversity in the information society. In SALTMIL2009 Workshop: IR-IE-LRL. Information Retrieval and Information Extraction for Less Resourced Languages. University of the Basque Country, Biscay, Spain.

[3] András Kornai. 2013. Digital language death. PloS one, 8, 10 (October 2013), 1-11.

[4] Steven Krauwer. 2003. The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. In Proceedings of SPECOM 2003. SPECOM, Moscow, 8-15.

[5] José I. Hualde and Koldo Zuazo. 2007. The standardization of the Basque language. Language Problems and Language Planning, 31, 2 (2007), 143-168.

[6] Irene López-Goñi. 2003. Ikastola in the twentieth century: an alternative for schooling in the Basque Country. History of Education, 32, 6 (2003), 661-676.

[7] Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra and Ivan Meza-Ruiz. 2018. Challenges of language technologies for the indigenous languages of the Americas. In Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 55-69. arXiv preprint arXiv:1806.04291.

[8] Hossein Hassani. 2018. BLARK for multi-dialect languages: towards the Kurdish BLARK. Language Resources and Evaluation, 52, 2 (June 2018), 625-644.

[9] Oliver Streiter, Kevin P. Scannell and Mathias Stuflesser. 2006. Implementing NLP projects for non-central languages: instructions for funding bodies, strategies for developers. Machine Translation, 20, 4 (March 2006), 267-289.

[10] Mikel L. Forcada. 2006. Open source machine translation: an opportunity for minor languages. In Proceedings of the Workshop Strategies for developing machine translation for minority languages LREC (Vol. 6). LREC, Genoa, Italy, 1-6.

[11] Robert Munro. 2010. Crowdsourced translation for emergency response in Haiti: the global collaboration of local knowledge. In proceedings of the AMTA Workshop on Collaborative Crowdsourcing for Translation. Association for Machine Translation in the Americas, Denver, Colorado, 1-4.

[12] Marta Sabou, Kalina Bontcheva and Arno Scharl. 2012.. Crowdsourcing research opportunities: lessons from natural language processing. In Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies. Association for Computing Machinery, New York, NY, United States, 1-8.

[13] Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. Computational linguistics, 29, 3 (September 2003), 333-347.

[14] Igor Leturia, Antton Gurrutxaga, Iñaki Alegria, and Aitzol Ezeiza. 2007. CorpEus, a 'web as corpus' tool designed for the agglutinative nature of Basque. In Building and exploring web corpora, Proceedings of the 3rd Web as Corpus workshop. Presses universitaires de Louvain, France, 69-81.

[15] Nerea Areta, Antton Gurrutxaga, Igor Leturia, Iñaki Alegria, Xabier Artola, Arantza Diaz de Ilarraza, Nerea Ezeiza and Aitor Sologaistoa. 2007. ZT Corpus: Annotation and tools for Basque corpora. In Proceedings of Corpus Linguistics 2007. UCREL, Lancaster, 1-19.

[16] Ximena Gutierrez-Vasques, Gerardo Sierra and Isaac H. Pompa. 2016. Axolotl: a Web Accessible Parallel Corpus for Spanish-Nahuatl. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). European Language Resources Association (ELRA), Portorož, Slovenia, 4210–4214.

[17] Steven Abney and Steven Bird. 2010. The human language project: building a Universal Corpus of the world's languages. In Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Uppsala, Sweden, 88-97.

[18] Kevin P Scannell. 2007. The Crúbadán Project: Corpus building for under-resourced languages. In Building and Exploring Web Corpera: Proceedings of the 3rd Web as Corpus Workshop (4). Presses universitaires de Louvain, Louvain-la-Neuve, Belgium, 5-15.

[19] Eloína Miyares Bermúdez, Leonel Ruiz Miyares, Cristina Álamo Suárez, Celia Pérez Marqués, Xabier Artola Zubillaga, Iñaki Alegría Loinaz and Xabier Arregi Iparragirre. 2010. La segunda y tercera ediciones del Diccionario Básico Escolar. In Proceedings of the 14th EURALEX International Congress. Fryske Akademy, Leeuwarden/Ljouwert, The Netherlands, 519-526.

[20] Mans Hulden. 2009. Foma: a finite-state compiler and library. In Proceedings of the 12th Conference of the EACL: Demonstrations Session. Association for Computational Linguistics, Athens, Greece, 29-32.

[21] Iñaki Alegria, Izaskun Etxeberria, Mans Hulden and Montserrat Maritxalar. 2009. Porting Basque morphological grammars to foma, an open-source tool. In International Workshop on Finite-State Methods and Natural Language Processing. Springer, Berlin 105-113.

[22] Annette Rios. 2011. Spell checking an agglutinative language: Quechua. In 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. LTC, Poznań, Poland, 51-55.

[23] Rodrigo Agerri, Josu Bermudez and German Rigau. 2014. IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In Proceedings of the 9th Language Resources and Evaluation Conference (LREC2014). LREC, Reykjavik, Iceland, 3823-3828.

[24] Olatz Perez-de-Viñaspre and Maite Oronoz. 2015. SNOMED CT in a language isolate: an algorithm for a semiautomatic translation. BMC Medical Informatics and Decision Making, 15, Suppl 2:S5. doi:10.1186/1472-6947-15-S2-S5

[25] Olatz Perez-de-Viñaspre a Gorka Labaka. 2016. IXA Biomedical Translation System at WMT16 Biomedical Translation Task. In Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers. Association for Computational Linguistics, Berlin, Germany, 477–482.

[26] Xabier Soto, Olatz Perez-De-Viñaspre, Maite Oronoz and Gorka Labaka. 2019. Leveraging SNOMED CT terms and relations for machine translation of clinical texts from Basque to Spanish. In Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation. European Association for Machine Translation, Dublin, Ireland, 8-18.

[27] Xabier Soto, Olatz Perez-de-Viñaspre, Gorka Labaka and Maite Oronoz. 2019. Neural Machine Translation of clinical texts between long distance languages. JAMIA (Journal of the American Medical Informatics Association), 26, 12 (December 2019), 1478–1487. https://doi.org/10.1093/jamia/ocz110

[28] Mikel Artetxe, Gorka Labaka and Eneko Agirre. 2019. Bilingual lexicon induction through unsupervised machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 5002-5007.

[29] Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, 984-996.

[30] Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In Proceedings of the 2017 conference on empirical methods in natural language processing. Association for Computational Linguistics, Copenhagen, Denmark, 2832-2838.

[31] Mikel Artetxe, Gorka Labaka, Eneko Agirre and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In Proceedings of the Sixth International Conference on Learning Representations. ICLR, Vancouver, Canada.

[32] Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka and Eneko Agirre. 2020. A Call for More Rigor in Unsupervised Cross lingual Learning. In Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online. https://arxiv.org/pdf/2004.14958.pdf

[33] Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa and Eneko Agirre. 2020. Give your Text Representation Models some Love: the Case for Basque. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020). LREC, Marseille, France, 4781-4788.

[34] Arantxa Otegi, Aitor Agirre, Jon Ander Campos, Aitor Soroa and Eneko Agirre. 2020. Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque. In Proceedings of The 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, 429–435.

A Closer Look at Welsh Word Embeddings

GERAINT PALMER

School of Mathematics, Cardiff University, Wales

PADRAIG CORCORAN

School of Computer Science and Informatics, Cardiff University, Wales

LAURA ARMAN

School of English, Communication and Philosophy, Cardiff University, Wales

DAWN KNIGHT

School of English, Communication and Philosophy, Cardiff University, Wales

IRENA SPASIC

School of Computer Science and Informatics, Cardiff University, Wales

This chapter presents a set of general word embeddings for the Welsh language. These are vector representations of words in language that reflect their semantic meaning in mathematical form, and are able to support a range of natural language processing tasks, and in particular deep learning methods. We discuss the quality of the training corpus and the embeddings that were automatically learnt from the corpus. We use qualitative analysis to identify any potential improvements to the embeddings that are specific to the Welsh language.

Keywords: word embeddings, language technologies, the Welsh language

1 INTRODUCTION

This chapter presents work on the training of a set of Welsh language word embeddings, which can support several natural language processing tasks. Our aim is to create a comprehensive set of word embeddings that contain information on the syntax and semantics of the Welsh language, to improve and help develop natural language processing models for the Welsh language. In this chapter we provide some background, we discuss the importance and quality of the training corpus, and look for clues to ascertain the quality of the embeddings we create by examining their vector space. This type of analysis is important because any weaknesses and biases in the embeddings can be inherited by the language technologies that use them.

Section 2 provides a theoretical background, with Section 2.1 describing what word embeddings are, Section 2.2 describing their potential use, and Section 2.3 describing the training process. Section 3 then describes the work, with Section 3.1 describing the corpus, and Section 3.2 discussing aspects of the quality of the word embeddings.

2 BACKGROUND TO WORD EMBEDDINGS

This section provides a theoretical background to word embeddings, what word embeddings are, their potential use, and how they are trained.

2.1 What are Word Embeddings?

Word embeddings are a mapping of lexico-semantic space to real-value *n*-dimension vector space, $\mathbb{R}^n$ [1]. That is, a mapping of words in a language to vectors of real numbers. We can think of this so that every word in a language has a real vector to represent it.

Good quality word embeddings have specific properties relating to the meaning and normal context of words. In other words, the aim is for the vectors to summarise analytical semantic information. For the mapping to store relevant semantic information, multi-dimensional vectors are used, usually with $100 \leq n \leq 500$. Ideally, relationships between the vectors of the embeddings should reflect the semantic relationships between their corresponding words. If $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^n$, where $\mathbf{x}_1$ is the vector representing the word "llew" (lion), $\mathbf{x}_2$ is the vector representing the word "teigr" (tiger), and $\mathbf{x}_3$ is the vector representing the word "llyfr" (book), then $||\mathbf{x}_1 - \mathbf{x}_2||$ should be significantly less than $||\mathbf{x}_1 - \mathbf{x}_3||$, where the operation $||\cdot||$ is a norm representing distances in *n*-dimension. That is, the vectors representing "llew" and "teigr" should be much closer together than the vectors representing "llew" and "llyfr", because they are more connected. Figure 1 shows a two-dimensional representation of this idea, with similar words closer together than dissimilar words.

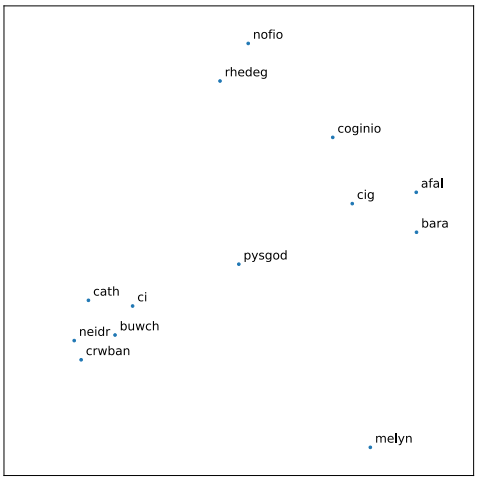

Figure 1: A two-dimensional example of the relationship of word embeddings to semantic meaning.

Another useful property of word embeddings is that the size and direction of the difference between vectors is similar if and only if the semantic differences between their corresponding words are broadly equal. For example, if $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d \in \mathbb{R}^n$, and $\mathbf{x}_a$ is the vector representing the word "brenin" (king), $\mathbf{x}_b$ is the vector representing the word "brenhines" (queen), $\mathbf{x}_c$ is the vector representing the word "athro" (male teacher), and $\mathbf{x}_d$ is the vector representing the word "athrawes" (female teacher); then travelling from $\mathbf{x}_a$ to $\mathbf{x}_b$ should be similar to travelling from $\mathbf{x}_c$ to $\mathbf{x}_d$, because the pairs of words "brenin" and "brenhines" and "athro" and "athrawes" have the same relationship with each other. In terms of vector addition properties, this means that $\mathbf{x}_a = \mathbf{x}_c - \mathbf{x}_d + \mathbf{x}_b$, and this is shown in two-dimensional form in Figure 2.

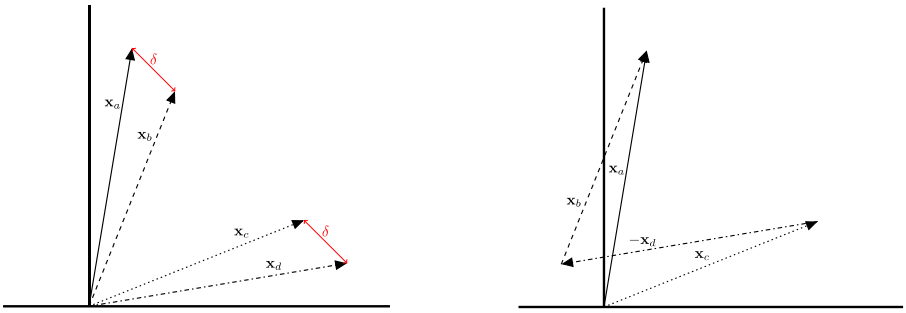

Figure 1: An example of the addition of vectors and their relationship to equivalent semantic meanings.

The fact that we can treat words as mathematical objects, along with the critical attributes described above, means that they can model logic around natural language using simple vector operations often used in deep learning algorithms. Some applications of word embeddings are described in the next section.

2.2 Use of Word Embeddings

Word embeddings have been proven to improve the performance of a number of natural language processing models and tasks. Using a set of vectors, which includes semantic connections between words, in natural language processing tasks, means that they perform well on the out-of-vocabulary problem. That is they can make predictions on words they have not seen before (i.e. in the training process), by taking advantage of their links to other words in the training data. The type of tasks includes:

- *Machine translation:* the use of word embeddings in neural machine translation tasks has proved to improve their performance, for example in [2].
- *Sentiment analysis:* this is a task to categorize sentences by positive, neutral, or negative sentiments. This has applications in areas such as commercial research and analysis of social media trends. [3] is an example of where word embeddings improve performance.
- *Named entity recognition:* this is a task that aims to identify phrases that act as names, for example "prif weinidog" (*prime minister*) or "cancr y fron" (*breast cancer*). This has applications in searching databases from corpora, particularly in academic and medical areas. [4] is an example of word embeddings used for this purpose.
- *Dependency parsing:* a task to analyse sentence structure through word labels within a sentence such as parts of speech (names, verbs, adjectives etc) and to identify the links between them. For example [5] in the sentence "gwelodd Sian y ci" (*Sian saw the dog*) the verb "gwelodd" (*saw*) links two names, the subject "Sian" and the object "ci" (*dog*). [6] is an example of using word embeddings to do this.

The Welsh language is described as a low resource language in terms of natural language processing. This means that the availability of data and corpora is limited compared to languages spoken more commonly, and so training high performing natural language processing models can be a challenge. Word embeddings can therefore act as enablers of complex natural language processing applications.

2.3 Training Word Embeddings

Any mapping of lexical space to vector space is called word embeddings, but to be useful for other natural language processing tasks we want them to display certain properties, such as those discussed in Section 2.1. To achieve this, there are a number of ways to build word embeddings, and machine learning techniques are becoming increasingly

popular. In terms of machine learning, we *train* word embeddings from input data, namely a sufficiently large corpus of text.

In this work we looked at two methods of training word embeddings, *Word2Vec* ([7]) and *FastText* ([8]), which can use two different algorithms, *Skip-gram* ([9]) and *Continuous Bag of Words* (*CBOW*, [7]). These rely on the theory of distributional semantics, which claims that words appearing in the same context try to convey similar meanings [10]. So these techniques assume that the meaning of a word depends mainly on its context, that is the frequency of words that are close neighbours to it within sentences. *Word2Vec* takes this idea literally. Word spelling is not looked at, each word is treated as a separate n-gram, and the other n-grams that appear close to it most often within a sentence are considered. This can be useful for analytic languages such as Mandarin. *FastText* however also considers subwords (different components within written words), which are useful for languages with rich morphology, for example Turkish.

The aim of these algorithms is to identify vectors so that he total SoftMax distance between each n-gram and its neighbours reflects the likelihood of finding a word in some neighbourhood of other words. The neighbours of an n-gram are defined as the n-grams that are within a context window to it in a sentence, that is the k nearest neighbours. An example is shown in Figure 3. The *Skip-gram* algorithm seeks to reflect the likelihood of that word appearing subject to the context, while *CBOW* seeks to reflect the likelihood of the context appearing subject to that word. Their performances therefore vary between the more and less frequent words in the training corpus.

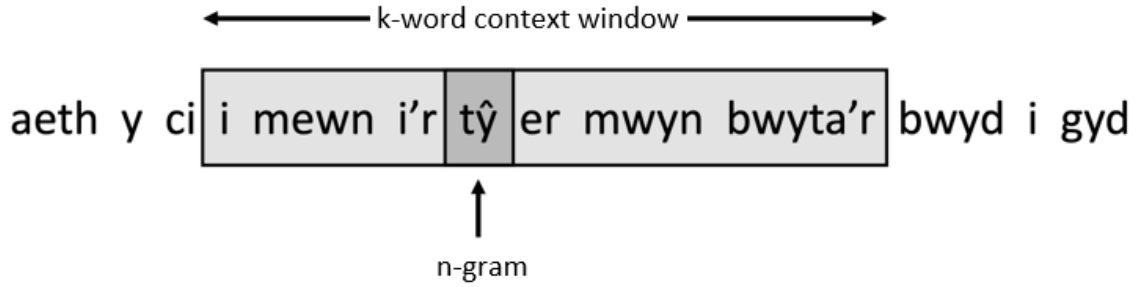

Figure 3: An example of a word within its context window.

3 WELSH WORD EMBEDDINGS

We trained a set of word embeddings for the Welsh language, as described in [11]. We collected a corpus, described in Section 3.1, then trained embeddings using each combination of the techniques and algorithms described in Section 2.3. We evaluated these embeddings using a range of quantitative techniques ([12], [13]), and the results are reported in [11]. Section 3.2 qualitatively evaluates the set of embeddings that performed best on these quantitative techniques.

3.1 Corpus Collection

As described in Section 2.3, word embeddings need to be trained on a corpus of Welsh language documents, and the properties of the corpus greatly influence the properties of the word embeddings. For commonly spoken languages, such as English, the availability of corpora is not a significant problem. For example Wikipedia, which

currently contains over 3.7 billion words, is used to train English models. Because the Welsh language has fewer speakers and fewer resources, the task of collecting such a corpus for Welsh is more challenging.

A number of small, specialist corpora are available for the Welsh language, all too small to be useful on their own. So we collected a larger corpus by combining a number of these Welsh language resources, as well as using web scraping tools to access additional texts. Another consideration was the variety of the corpus, as natural language processing models trained on specialist corpora are only useful for tasks in specialist domains. Because our task was to create a set of generic word embeddings for use in general natural language processing tasks, we tried to collect a sufficiently varied corpus. In all, our corpus consists of 92,963,671 words. Table 1 shows the sources and sizes of data used. We took care to ensure that the sources did not contain the same original text, and included a variety of different subjects and language registers.

The Bible is often used as the first port of call when building a corpus as it has been translated into a large number of languages and is usually available free of charge in electronic form, but it has disadvantages including archaic, formal language, and the limited diversity of its subject matter. The corpus An Crúbadán is a large collection of websites, blogs, and Welsh tweets. We received an introductory subset of the electronic corpus CorCenCC, which includes emails and text messages. Together these provide a wide range of subjects in contemporary, possibly informal, Welsh. Gwerddon is a multi-disciplined academic journal in Welsh; and the DECHE project includes academic textbooks in Welsh. These corpora provide samples of highly formal academic writing in a range of specialist subjects. Google Corpuscrawler is an open source tool which can be used to access corpora in many different languages. In running the tool for the Welsh language, it mainly accesses the articles of BBC Cymru Fyw. To add to the ready-made sources, we also accessed other magazine and news websites, Golwg360, O'r Pedwar Gwynt, Barn, and PoblCaerdydd, which provide examples of journalistic language. The proceedings of the National Assembly are available, of which two pre-existing corpora are available, one which includes the proceedings from 1999 to 2006, the other from 2007 to 2011, to ensure that formal spoken language on political topics is included. Cronfa Electroneg o Gymraeg (CEG, *The Electronic Corpus of Welsh*) includes examples of fictional prose and administrative documents which also add to the diversity of the language included in the corpus, and Welsh Wikipedia contains articles on a wide range of subjects, both general and specialist. This is the largest single source available free of charge in Welsh.

Table 1: Sub-corpora sources

Source	Number of words
The Bible[1]	749,573
An Crúbadán Corpus [14]	22,572,066
CorCenCC[2]	1,875,540
CEG [15]	1,046,800
Google Corpuscrawler[3]	14,791,835
Gwerddon[4]	767,677
DECHE Project [16]	2,126,153
Proceedings of the National Assembly 1999-2006 [17]	11,527,963
Proceedings of the National Assembly 2007-2011 [18]	8,883,970
Wicipedia Cymraeg[5]	21,233,177
Other miscellaneous websites[6]	7,388,917
Total	92,963,671

3.2 Evaluation

We used the corpus described in Section 3.1 to train a number of word embeddings with each combination of the techniques described in Section 2.3, *FastText* and *Word2Vec*, *Skip-gram* and *CBOW*, as well as a number of ways of tokenizing the corpus. We used a number of quantitative techniques to evaluate the word embeddings. The set of word embeddings that performed best quantitatively is available to download here: https://datainnovation.cardiff.ac.uk/is/wecy/.

It is also important to evaluate the word embeddings qualitatively, that is visually, to see how well the word embeddings represent word semantics independently of the automatic evaluation technique used. One standard way to do this is to choose a word, search for the set of other vectors closest to the vector representation of that word in the vector space, and check if the words represented by those vectors are semantically linked to the original word. This is similar to the automatic technique described in [19]. The advantage of doing this automatically is being able to explore more words more effectively, but the advantage of doing this visually is to be able to observe connections, similar features, and patterns that were not identified before starting. It also gives us the opportunity to analyse the nature of the connections, and to look further at the words that are not connected, which can give us an insight into the potential improvements of the model.

[1] http://www.beibl.net/

[2] https://www.corcencc.org/

[3] https://github.com/google/corpuscrawler

[4] http://www.gwerddon.cymru/cy/hafan/

[5] https://cy.wikipedia.org/wiki/Hafan

[6] https://golwg360.cymru/, https://pedwargwynt.cymru/, https://barn.cymru/ and https://poblcaerdydd.com/

Here we have chosen five words and looked at their nearest ten vectors using the cosine distance. Table 2 gives the words chosen and their nearest ten words. We will look at them in turn to see what they tell us about the quality and structure of the word embeddings.

Table 2: The nearest ten words of a sample of words.

Word	Nearest words
arian	harian, arian', ariannu, cyllid, bunnau, gyllid, bres, arianu, goffrau, wario
cysgu	gysgu, deffro, llewygu, cysgai, chwtsho, chysgu, dihuno, cerdded, ddihuno, gysgai
blawd	blawdog, flawd, blawr, blawdty, siaradblawd, menyn, fenyn, cyflasynnau, blaw, chwstard
tonyrefail	donyrefail, nhonyrefail, ffosyrefail, maesyrefail, trebanog, tonysguboriau, tonteg, tonypandy, trecynon, trealaw
actores	actor, actoresau, sgriptwraig, digrifwraig, berfformwraig, comediwraig, chantores, gantores, ddigrifwraig, perfformwraig

In turn:

- ***"arian" (*money*):*** All ten nearest words relate to the financial topic. This suggests that this model has the first desirable property discussed in Section 2.1. Note also that there are mutations and misspellings. For the word embeddings to be useful for natural language processing tasks, they need to describe the type of language on which the tasks will be used, and therefore the type of language people actually use. The inclusion of mutations and misspellings is therefore also desirable, as this is how people use Welsh in everyday life, the evidence of this is that these words appear often enough in our corpus to appear in the word embeddings.
- ***"cysgu" (sleep):*** The nearest ten words are all verbs of different forms, and broadly related.
- ***"blawd" (flour):*** The nearest ten words show related words and ingredients. There are three exceptions, "siaradblawd", "blawr", and "blaw". Further research indicates that the first is a tweeting hashtag.

  It appears that the other two, however, only appear as their spelling is similar to the original word, which happens because the *FastText* algorithm considers sub-words as n-grams. This is inconvenient, and a sign of a corpus that is too small. We also note that the synonym "fflŵr" does not appear, perhaps because this word appears more often in the spoken language than in written language, and what we have is mainly a written corpus.

- ***"tonyrefail":*** Apart from mutations, all words in the nearest ten words are also towns. Furthermore, each is a town in south Wales, very close geographically to Tonyrefail. This phenomenon has been observed with a number of small and medium sized towns in Wales, perhaps because neighbouring towns are mentioned in the same news articles that form part of our corpus.
- ***"actores" (actress):*** All ten nearest words relate to the entertainment industry. Apart from the first two words, which are different forms of the original word, each word is in its female form. Whilst the property of understanding the forms of different words is desirable for good quality word embeddings, this may suggest that the model has differentiated between the female and male forms of the jobs rather than the nature of the jobs themselves. This is not evidence that this has happened, but it needs to be noted that by training future language technologies on past data, social stereotypes can be reinforced by the model. This is a well-known problem, where word embeddings can show racist and sexist biases due to the training data [20].

4 CONCLUSIONS

Overall, the qualitative evaluation of these word embeddings agrees with the quantitative evaluation that they are performing adequately, that is as they are expected to do in terms of reflecting language syntax and semantics. Looking more closely at anomalies has helped us to better understand the vectors, to acknowledge any potential weaknesses, to explore their potential causes, and to explore ways in which we can prevent these weaknesses being inherited by the natural language processing tasks that will use them.

The word embeddings described in this chapter are available as open source from the following website: https://datainnovation.cardiff.ac.uk/is/wecy/, together with a tool to find the nearest ten words of any word in the vector space.

ACKNOWLEDGEMENTS

This study was funded by the Welsh Government as part of a project linked to the Welsh language processing infrastructure. We would like to thank Gareth Morlais, CorCenCC, and Kevin Scannell for their support in collecting and evaluating the corpus.

REFERENCES

[1] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine 13, 3 (August 2018), 55-75.

[2] Ye Qi, Devandra Singh Sachan, Matthieu Felix, Sarguna Janani Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Association for Computational Linguistics, New Orleans, Louisiana, 529-535.

[3] Maria Giatsoglau, Manolis G Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, and Konstantinos Ch Chatzisavvas. 2017. Sentiment analysis leveraging emotions and word embeddings. Expert Systems with Applications 69 (March 2017), 214-224.

[4] Yonghui Wu, Jun Xu, Min Jiang, Yaoyun Zhang, and Hua Xu. 2015. A study of neural word embeddings for named entity recognition in clinical text. In AMIA Annual Symposium Proceedings 2015. AIMA, San Francisco, CA, 1326-1333.

[5] Dewi Bryn Jones, Delyth Prys, Myfyr Prys, and Gruffudd Prys. 2019. Llawlyfr technolegau iaith. Coleg Cymraeg Cenedlaethol. Bangor, Wales.

[6] Timothy Dozat, and Christopher D Manning. 2018. More accurate semantic dependancy parsing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 2 (Short Papers). Association for Computational Linguistics, Melbourne, Australia, 484-490.

[7] Tomas Miklov, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In Proceedings of the International Conference on Learning Representations (2013). ICLR, Scottsdale, AZ, USA.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomos Mikolov. 2017. Enriching word sectors with subword information. Transactions of the Association for Computational Linguistics 5 (2017), 135-146.

[9] Tomos Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In Neural Information Processing Systems. Annual Conference. 27th 2013. (4 Vols) Advances In Neural Information Processing Systems 26. NIPS, Lake Tahoe, Nevada, USA, 3136-3144.

[10] Zellig S Harris. 1954. Distributional structure. WORD 10 2-3 (1954), 146-162.

[11] Padraoig Corcoran, Geraint Palmer, Laura Arman, Dawn Knight, Irena Spacis. 2021. Creating Welsh Language Word Embeddings. Applied Sciences 11(15) 6896. https://www.mdpi.com/2076-3417/11/15/6896

[12] Tobias Schnabel, Igor Labutov, David Mismo, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, 298-307.

[13] Amir Bakarov. 2018. A survey of word embeddings evaluation methods. arXiv preprint arXiv:1801.09536.

[14] Kevin P Scannell. 2007. The Crúbadán Project: Corpus building for under-resourced languages. In Building and Exploring Web Corpera: Proceedings of the 3rd Web as Corpus Workshop (4). Presses universitaires de Louvain, Louvain-la-Neuve, Belgium, 5-15.

[15] N Ellis, C O'Dochartaigh, W Hicks, M Morgan, and M Laporte. 2001. Cronfa Electroneg o Gymraeg (CEG): A 1 million word lexical database and frequency count for Welsh.

[16] Delyth Prys, Mared Roberts, and Dewi Bryn Jones. 2014. DECHE and the Welsh national corpus portal. In Proceedings of the First Celtic Language Technology Workshop. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 71-75.

[17] Dafydd Jones, and Andreas Eisele. 2006. Phrase based statistical machine translation between English and Welsh. In Proceedings of the 5th SALTMIL Workshop on Minority Languages at the 5th International Conference on Language Resources and Evaluation. LREC, Genoa, Italy,

75-77.

[18] Kevin Donnelly. 2013. Kynulliad3: a corpus of 350,000 aligned Welsh and English sentences from the Third Assembly (2007-2011) of the National Assembly for Wales. http://cymraeg.org.uk/kynulliad3

[19] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. ACM Transactions on Information Systems 20, 1 (January 2002), 116-131.

[20] James Zou, and Londa Schiebinger. 2018. AI can be sexist and racist - it's time to make it fair. Nature 559, 7714 (July 2018), 324-326.

A practical implementation of a Porter stemmer for Welsh

VIGNESHWARAN MURALIDARAN

School of Computer Science & Informatics, Cardiff University, Wales

GERAINT PALMER

School of Mathematics, Cardiff University, Wales

LAURA ARMAN

School of English, Communication and Philosophy, Cardiff University, Wales

KEZIAH O'HARE

School of English, Communication and Philosophy, Cardiff University, Wales

DAWN KNIGHT

School of English, Communication and Philosophy, Cardiff University, Wales

IRENA SPASIĆ

School of Computer Science & Informatics, Cardiff University, Wales

Keywords and Phrases: Natural language processing, Information retrieval, Stemming, Lemmatization

1 INTRODUCTION

Stemming is the process of reducing a word to its stem. A stem is the base form of a word from which new words can be created by attaching affixes through processes of inflection and derivation. For example, words such as *acts*, *active*, *actions*, *activated*, *activation*, *activity*, *actor*, *actress* would all be stemmed to the base form *act*. Similarly, words such as *move*, *movement*, *mover*, *moving*, *moves*, *movie* would all be stemmed to the base form *mov*. In most cases (e.g. *act*) with a few exceptions (e.g. *mov*), the stem will be identical to the morphological root of a word. The root word is the primary lexical unit of a word that carries the most significant semantic content. It is also atomic in the sense that it cannot be decomposed to smaller constituents. Therefore, a stem as a proxy for a morphological root will reflect the essence of the meaning associated with a whole family of related words (see examples above). This is what makes stemming valuable to natural language processing (NLP) applications as it facilitates semantic analysis without having to resort to lexical resources such as thesauri, which are not always readily available, and may be relatively expensive to develop.

Stemming algorithms such as the Porter stemmer [1] are light-weight practical solutions to neutralising inflection and derivation, hence focusing on the core meaning conveyed. The algorithm strips off common affixes (prefixes and suffixes) iteratively while addressing the morphological changes that may affect the surface form. For example, when the suffix –ness is combined with the word *happy*, the vowel *y* changes into *i*, i.e. *happy* + *–ness* = *happiness*. Therefore, working backwards, when stripping off the suffix *–ness* from *happiness*, the vowel *i* in *happi* needs to be reverted to *y*. Handling such interactions within the individual stemming rules requires a careful

studying of the language morphology and this step requires linguistic expertise. Rules can be developed to remove inflectional morphemes from a word such as noun declensions or verb conjugations. Given two words W1 and W2, there will always be a difference of opinion on whether they should be conflated into a common stem or not. For instance, the words *related* and relativity can be reasonably conflated to the single stem *relat*. However, if the corpus in question contains documents related to Physics it is a mistake to conflate these terms into one base form because the word *relativity* itself is a concept with special meaning which should not be conflated with other words. Even if the success rate of a stemmer is less than 100% due to such problems, developing a simple and fast algorithm that strips away the most common suffixes typically improves the performance of information retrieval.

2 RELATED WORK

Word stemming is used as a pre-processing step in many information retrieval techniques especially search and indexing algorithms. By conflating several variants of a word into a single stem, a search algorithm can treat all the variants of a word as relevant for the search results and this improves the performance of the search algorithm. Lemmatization is a similar technique, which also reduces different word forms to a single base form called a lemma. A lemmatizer recognises the base form by considering the grammatical category of the word, often by taking into account the context of its occurrence, and then derives the lemma based on this information. For example *do*, *did*, *done* would all be conflated to *do* after lemmatization because they belong to the same part of speech and are grammatical variants of the same verb. On the other hand, a stemmer tries to conflate variants into a single base form without considering grammatical categories and context. Another important difference is that the base form obtained by the lemmatizer should be a dictionary form or morphological base form of a word. This is not the case with the stemmer output. Thus *reduce*, *reduction*, *reduced* are conflated into *reduc* by a stemmer while a lemmatizer would identify *reduce* as the base form.

Stemming implementations can be grouped into truncating methods, statistical methods and mixed methods [2]. In English and several other languages morphological variations occur at the end of a word form [3]. This has given rise to truncating algorithms that employ user-defined suffix truncation rules to obtain the stems. Truncating algorithms retain n letters of a word and strip off the letters after the n-th position. S-stemmer [4] and Lovins stemmer [5] are examples of suffix-truncation algorithms. Porters stemmer [1], Porters2 (or Snowball stemmer) [6], [7] and the Lancaster stemming algorithm [8] are the major stemming algorithms that strip the derivational and inflectional suffixes off a word to get the base form. There are other implementations of stemmers that perform statistical analysis on a corpus of words to identify the group of words that should be conflated to the same base form. In an n-gram stemmer every word is seen to be made of a sequence of characters. By extracting character n-grams for the words in a corpus, it is possible to conflate variants of the same base form because similar words will have a high number of n-grams in common.

Melucci and Orio [9] proposed an unsupervised approach to stemming using the Hidden Markov Model where the sequence of letters in a word are modelled as concatenation of stem set and suffix set. A word is built as a result of transition between these states. Although this method is language-independent it is complex and overstems the words sometimes. There are implementations, such as Krovetz Stemmer [10] and Xerox stemmer [11], which are a mixture of corpus-based methods along with the analysis of inflectional and derivational morphology. Xu and Croft [12] proposed corpus-based stemming using cooccurrence of word variants. In corpus-based methods, the words that have a common stem are modified to suit the characteristics of a given text corpus. The basic idea is that variants of a stem should co-occur in the documents from a corpus. Funchun Peng et al. [13] proposed a context-

sensitive stemmer intended to be applied for web search queries. The basic idea is to predict useful variations of a word using some other stemmer such as the Porter stemmer and context-sensitive matching is done to these variants to improve the quality of search results.

Each of these approaches have their own advantages and disadvantages. While none of them perform stemming with perfect accuracy, the choice of the most appropriate approach depends on the language in question, the target application, and the availability of linguistic resources such as lexica and the expertise provided by a language specialist. Out of all the approaches described above, the original Porters stemmer algorithm proposed for English is the most widely adapted stemmer for other languages. Being a light approach to stemming it does not require any complex data structures or processing strategies. The implementation methodology is described in section 3.

3 METHODOLOGY

We implemented a stemming algorithm for the Welsh language by adapting the principles of the Porter stemmer algorithm [1], which was originally implemented for English, but has since been re-implemented for a large family of European languages [7]. It is an iterative, rule-based algorithm that defines explicitly a set of suffixes (the criterion under which the suffix stripping rules are applied) and the scope and order of application of rules. The key to producing a good stemming algorithm for a new language is creating an appropriate rule base and the right order in which the rules should be applied. In this project, a team of linguists (also native Welsh speakers) developed a fine-grained list of stemming rules. In comparison to English, where inflection can be dealt with by focusing on a single word at a time without considering the neighbouring words, this context-free approach is not suitable for Welsh, where some morphological changes (e.g. mutation) are context sensitive. To effectively address this problem, we integrated lemmatization into the stemming process. Specifically, we lemmatize each word, i.e. map it to their canonical form (e.g. singular for nouns, infinitive for verbs, etc.), prior to applying stemming rules. This solves the "word interaction" problem and allows each word to be stemmed independently of its context. In practical terms, it reduces the number of stemming rules as well as the complexity of the ways in which they can be combined, making the stemming algorithm more robust and easier to maintain. Lemmatization in our approach is performed by CyTag, a part-of-speech tagger for Welsh developed as part of the CorCenCC project [14].

The Welsh stemming algorithm can be summarised using the following list of high-level rule descriptions applied in the given order:

1. Identify a set consonants *C* and set vowels *V* that make up a syllable in the Welsh language. Given a word to be stemmed, find the measure of words *m* which is defined as the number of times the sequence *VC* occurs in a word.
2. Identify a set of mutation rules. Remove the mutation if applicable. If multiple outcomes are possible, then perform disambiguation.
3. Remove the gender-number derivative and inflection suffixes from a word to derive noun and verb stems.
4. Remove comparative and superlative suffixes to get adjective stems.
5. Identify a list of productive suffixes that derive nouns, verbs, adjectives and adverbs from a basic form, e.g. *iau/au/od/ed/i/yn/ion/ydd/oedd*. Remove these suffixes from a word.
6. For the above steps 3-5, make use of the measure *m* to identify if the rule is applicable for a word or not, e.g. the word *tad* is not reduced as *t* by removing the suffix *-ad* because its measure is not greater than 1.

7. Remove productive prefixes from the word.

The following subsections provide more information on the stemming rules.

3.1 Definitions

In order to describe the algorithm, the following definitions are needed.

- **Consonants C** - A consonant in the Welsh language is formally defined as anything other than the letters *a*, *e*, *i*, *o*, *u*, *y* and the letter *w* with the following conditions. The letter *w* is a consonant when it is not nested between two consonants or when it is followed by a vowel. For example, in the word *dwr*, the letter *w* is a vowel whereas in the word *dwy*, the letter is a consonant. The Welsh language has eight consonant sounds which are written as digraphs (two letters) but are considered as single letters. These letters are: *ch*, *dd*, *ff*, *ng*, *ll*, *ph*, *rh*, *th*. The digraphs are themselves made of two letters each of which are treated as consonants according to our definition. These digraphs are treated as single letters while we apply stemming rules described in section 3.2.
- **Vowels V** - Anything which is not a consonant is a vowel.
- **Measure m** - Any word can be represented in the form [C]VC{m}[V], where C represents a sequence of consonants, *V* represents a sequence of vowels, VC{m} represents any sequence of vowels followed by a sequence of consonants repeated *m* number of times. Here, *m* is called the measure of a word. The following examples illustrate Welsh words of different measures *m*.
  m = 0 : y, rhiw, trwy, dau
  m = 1 : gwynt, bod, bwrdd, llyfr
  m = 2 : gwyntoedd, byddaf, llyfrau

Recognising the number of VC sequences within a word is useful to condition the application of affix stripping rules. For example, if we define a rule that the plural suffix *au* can be removed from the words with m>=2, it is applicable on the word *llyfrau* because they have a measure m=2. This results in valid a stem *llyfr*. However, the suffix *au* will not be stripped off *dau* because its m=0. Such rules can be arranged into different levels based on the scope of their application. As an example, suffix truncation rules are applied on a word before the application of the prefix removal rules.

3.2 Stemming Rules for the Welsh Language

The definitions of consonants and vowels are modified for the Welsh language, to handle *w* as a vowel or consonant depending on the context. Welsh is morphologically richer than English. Moreover words in Welsh have mutation rules based on the preceding words, grammatical gender of the words and their grammatical function. In order to simplify the implementation of the stemmer, we used a Welsh lemmatizer that forms a component of CyTagger[2], which is a part-of-speech (POS) tagger developed for the Welsh language. The lemmatizer was used to map a word to its canonical form before applying the stemming rules. Wherever the lemma was available from the lemmatizer output, it was used to simplify the stemming rules. The stemming rules are applied in two steps: Step 1 contains suffix rules which are applied first. Step 2 contains prefix and mutation rules which are applied next. The Welsh stemmer repository is available on Github.[3]

[2]https://github.com/CorCenCC/CyTag

[3]https://github.com/CorCenCC/WelshStemmer

In step 1, the gender-number suffixes are removed from a word to get noun and verb stems, comparative and superlative suffixes are removed to get adjective stems and a list of productive suffixes that derive nouns, verbs, adjectives and adverbs from a basic form are removed e.g. *iau/au/od/ed/i/yn/ion/ydd/oedd*. In step 2, a list of productive prefixes and mutation rules are applied. e.g. *cyd-*, *gwrth-*, *hunan-*, *rhag-*, *di-*, *an-*, *ad-*, *cyf-* etc. A detailed list of prefix, suffix and mutation rules are described in the following sections. The measure m of a word is just one of the ways to determine whether a rule is applicable in a context or not. Some of the ways to determine the scope of application of the stemming rules are:

- whether a word starts with a vowel
- whether a word starts with a consonant
- whether a word contains a vowel
- whether a word is mutated or not
- the measure m of a word

The input text is pre-processed through CyTagger to get the lemmas, POS tags and grammatical numbers of the input words. The input word and the other information are processed through steps 1 and 2 as described above. The different types of rules applied in order are shown in the subsections to follow. In the following subsections SM refers to Soft Mutation (Treiglad Meddal), NM refers to Nasal Mutation (Treiglad Trwynol) and AM means Aspirate Mutation (Treiglad Llaes).

3.2.1 Removing Suffixes

3.2.1.1 Adjectives derived from nouns and verbs

By removing the suffixes from adjectives that are derived from nouns and verbs, we get the original noun or verb stems. Examples for this type of rules are given below.

-(i)ol

anobeithiol	hopeless	(an + gobaith hope [SM])
beirniadol	critical	(beirniad CRITIC)
cydwybodol	conscientious	(cydwybod conscience)
gogleddol	notherly	(gogledd north)
ieithyddol	linguistic	(ieithydd linguist)

-aidd

deuaidd	binary	(dau two)
oeraidd	wintry	(oer winter)
niwlaidd	fuzzy, blurred	(niwl haze,fog)

A noun attached to another noun within a compound word causes soft mutation (SM). This must be taken care of when identifying the stem.

llys (court) + mam (mother) = llysfam

3.2.1.2 Number-gender declension suffixes on a noun stem

The suffixes denoting gender and number are removed to get the noun stem. The most common ones are: *IAU/AU/OD/ED/I/YN/ION/YDD/OEDD*.

The suffix -ACH is not removed from adjectives as they are already a stem.

EXCEPTIONS: RHAGORACH -> RHAGOR
CYFRINACH -> CYFRIN

3.2.1.3 Superlative - adjective + -ACH

Suffixes indicating the comparative and superlative degrees of an adjective are removed to get the base form of the adjective.

E.G. MWYACH -> MWY
EXCEPTIONS: HYTRACH, CHWAETHACH

If the word ending in -ACH is a NOUN, this need not be removed. They are already a stem. The following exceptions were identified for this rule.

EXCEPTIONS:

CLOGYRNACH	-> CLOGYRN
CYFEDDACH	-> CYFEDD
CYFEILLACH	-> CYFEILL
CHWANTACH	-> CHWANT
CREPACH	-> CREP
CRWBACH	-> CRWB
CRWMAC	-> CRWM
GWYACH	-> GWY
LLINACH	-> LLIN
LLOSGACH	-> LLOSG
SIMACH	-> SIM
SINACH	-> SIN
SOTHACH	-> SOTH
SWBACH	-> SWB
TOLACH	-> TOL

Otherwise, the suffix can be removed.

-ACH	PETHEUACH	-> PETHEU
	YSBLEDDACH	-> YSBLEDD
	SIAFFLACH	-> SIAFFL
	PLANTACH	-> PLANT
	MERCHETACH	-> MERCHET
-IACH	BLEWIACH	-> BLEW
	HOLLIACH	-> HOLL

EXCEPTIONS:

	AFIACH	-> AFIACH

-AETH TAKE OFF ONLY FOR NOUNS AND VERB-NOUNS

	MEDDYGAETH	-> MEDDYG
	ALAETH	-> ALAETH

IMPORTANT -AETH/-IAETH

It is important to strip off the suffix **-IAETH** before removing the suffix **-AETH.** The order of application of rules ensures that the suffix **-IAETH** is first accounted for and taken away before removing **-AETH**.

-IAETH is taken off for nouns only

E.G.	CYFRIFIAETH	-> CYFRIF
	MILWRIAETH	-> MILWR

3.2.1.4 Other Suffixes

-AS	PRIODAS	-> PRIOD
-EB	TYSTEB	-> TYST
-EG	CYMRAEG	-> CYMRA

-ELL *Diminutive Suffixes*

BRIWELL	-> BRIW	CRAFELL-> CRAF
BOTYMELL	-> BOTYM	BYSELL -> BYS
TERFYNELL	-> TERFYN	

Removing the plural suffix -AU results in a singular stem. This is always applied on plural nouns or adjectives.

E.G. AMLINELLAU -> AMLINELL

On singular adjectives and verbs, this suffix is not taken off. There are a few exceptions to this.

EXCEPTIONS:
AMLHAU -> AML, AGOSÁU -> AGOS, ANNETHAU -> ANNETH, DADLAU -> DADL (THIS WORD CAN ALSO BE A NOUN, DEPENDS ON CONTEXT), DETHAU -> DETH

However, on singular nouns, the suffix is taken off.

-EN	SEREN -> SER, COEDEN -> COED, GEWNYNEN -> GWENYN (NUMBER)
-ES	YSGRIFENYDDES -> YSGRIFENYDD, ARGLWYDDES -> ARGLWYDD, BRENHINES -> BRENIN (GENDER)
-ES	BUCHES -> BUCH
-FA	GRADDFA -> GRADD
-IN	CRIBIN -> CRIB
-OEDD	DANNOEDD -> DANN
-OR	YSGUBOR -> YSGUB
-RED	GWEITHRED -> GWEITH
-WRAIG	GWEITHWRAIG -> GWEITH

MASCULINE SUFFIXES

-AD	CARIAD	-> CARI
-ADUR	HOLIADUR	-> HOLI

-AI	MABWYSIADAI	-> MABWYSIAD
-AINT	HENAINT	-> HEN
-AWD	PELAWD	-> PEL
-AWDWR	CREAWDWR	-> CRE
-CYN	BRYNCYN	-> BRYN
-DEB	CYTUNDEB	-> CYTUN
-DER	CYFIAWNDER	-> CYFIAWN
-DID	CALONDID	-> CALON
-DOD	UFUDD-DOD	-> UFUDD
-DRA	BUANDRA	-> BUAN
-DWR	ADEILADWR	-> ADEILAD
-EDD	BRWDFRYDEDD	-> BRWDFYD
-FEL	OERFEL	-> OER
-I	CALEDI	-> CALED
-IANT	FFYNIANT	-> FFYN
-ID	RHYDDID	-> RHYDD
-INEB	GWARINEB	-> GWARIN
-MON	POSTMON	-> POST
-OD	RHYFEDDOD	-> RHYFEDD
-OL	GOLYGYDDOL	-> GOLYGYDD
-OR	PORTHOR	-> PORTH
-RWYDD	AEDDFEDRWYDD	-> AEDDFED
-WCH	AFLERWCH	-> AFLER
-WR	GARDDWR	-> GARDD
-YCH	EURYCH	-> EUR
-YD	CLEFYD,IECHYD	-> CLEF, -> IECH
-YDD	CYNORTHWYYDD	-> CYNORTHWY
-YN	ADERYN	-> ADER

NEITHER

-AID	CWPANAID	-> CWPAN

UNLESS IT IS -RHAID, -RAID IS ATTACHED TO ONE SYLLABLE (AS THESE ARE COMPOUNDS).

-AN	BLEIDDIAN, GWREIGAN	-> BLEIDD, -> GWREIG
-ED	COLLED, SYCHED	-> COLL, -> SYCH
-IG	MORWYNIG, PENDEFIG	-> MORWYN,-> PENDEF
-OD	DYRNOD	-> DYRN
-OG	CYMYDOG ,FFEDOG	-> CYMYD,-> FFED
-YLL	GWYNTYLL	-> GWYNT

PLURAL NOUNS

-OS	PLANTOS	-> PLANT

PLURAL AND SINGULAR

-ACH	DYNIONACH	-> DYNION (PL.)
-ACH	CORRACH	-> CORR
-ACH	CLINDDARACH	-> CLINDDAR,
	CYFFEILLACH	-> CYFFEILL (VERB-NOUN)

For words ending -NT when plural need -NN, everything after -NN were taken off and -NT is added.

CANNOEDD	-> CA	-> CANT
DANNEDD	-> DA	-> DANT
DIWYLLIANNAU	-> DIWYLLIA	-> DIWYLLIANT

Exceptions: GWYNT, PONT, RHIANT

3.2.2 Step 2 - Removing Prefixes

3.2.2.1 Nouns and verbs

The following derivational prefixes from nouns and verbs are removed. It should be noted that these prefixes may be spelt with or without a hyphen.

m > 0, cyd
cyd- co-,con-:

cydbwysedd	balance	pwysedd weight, pressure
cyd-ddigwyddiad	coincidence	digwydd happen
cydweithwyr	colleagues	gweithwyr workers
cydfyw	cohabit	byw live
cydfynd	accompany	mynd go

Cydym-
gwrth- anti-, counter-, against:

GWRTHBLAID	OPPOSITION PARTY	PLAID (POLITICAL) PARTY
GWRTHGYNHYRCHIOL	COUNTERPRODUCTIVE	CYNHYRCHIOL PRODUCTIVE

hunan- self

HUNANBARCH	SELF-ESTEEM	PARCH RESPECT
HUNANLADDIAD	SUICIDE	LLADD KILL

rhag- pre-, fore-

RHAGWELD	FORESEE	GWELD SEE
RHAGFARN	PREJUDICE	BARN JUDGMENT

ym- [Often meaning *self* or *each other*]

YMOLCHI	WASH(ONSELF)	GOLCHI WASH
YMLADD	FIGHT	LLADD KILL

3.2.2.2 Adjectives

AF- UN-:

AFRESYMOL	UNREASONABLE	RHESYMOL REASONABLE
AFLWYDDIANUS	UNSUCCESSFUL	LLWYDDIANUS SUCCESSFUL

DI- UN-, -LESS, WITHOUT:

DIDRAFFERTH	WITHOUT PROBLEMS	TRAFFERTH TROUBLE, PROBLEMS
DI-GYMRAEG	NON-WELSH SPEAKING	WELSHLESS
DIBAID	CEASELESS	PAID, PEIDIO CEASE

EXCEPTION: DIABETIG, DIACRONIG, DIACEN, DIAFAEL

- INFLECTIONAL PREFIXES THAT CAUSE INTERNAL SM

AF- UN-:

AFRESYMOL	UNREASONABLE	RHESYMOL REASONABLE
AFLWYDDIANUS	UNSUCCESSFUL	LLWYDDIANUS SUCCESSFUL

DI- UN-, -LESS, WITHOUT:

DIDRAFFERTH	WITHOUT PROBLEMS	TRAFFERTH TROUBLE, PROBLEMS
DI-GYMRAEG	NON-WELSH SPEAKING	WELSHLESS
DIBAID	CEASELESS	PAID, PEIDIO CEASE

EXCEPTION: DIABETIG, DIACRONIG, DIACEN, DIAFAEL

DERIVATIONAL PREFIXES THAT CAUSE INTERNAL NM

AN-

ANGHOFIO	FORGET + COFIO	REMEMBER

INFLECTIONAL PREFIXES THAT CAUSE INTERNAL NM

AN-

ANNHEBYG	UNLIKELY	+ TEBYG	LIKELY
ANNARLLENADWY	ILLEGIBLE	+ DARLLENADWY	LEGIBLE
ANGHYSON	INCONSISTENT	+ CYSON	CONSISTENT
AMHOSIB	IMPOSSIBLE	+ POSIB	POSSIBLE

Internal NM words are regarded as radical. This is the only case in Modern Welsh where the NM is consistently applied.

NOTE:

-N of AN- drops when mutating B- to M-, C- to NGH- or P- to MH-:

AMHENDANT (INDEFINITE)	AN + PENDANT (DEFINITE)

All radicals beginning with TR- cause one of the resulting N's to drop.

ANRHEFN (CHAOS) NOT *ANNRHEFN	AN + TREFN (ORDER)

The order in which the suffixes and prefixes are removed from the stem can be seen from the stemmer code available online. After removing the suffixes and prefixes, mutation rules are applied to get the original unmutated base form.

4 EVALUATION

The efficacy of stemming is often measured by the fractional reduction in index size achieved through stemming. Index compression factor (ICF) is calculated according to the following formula:

$$ICF = (W - S) / W$$

where w is the number of distinct words before stemming and s is the number of distinct stems [14].

We used CorCenCC [15], a national corpus of contemporary Welsh, and its computational infrastructure [16] to perform our experiments. The CorCenCC corpus was pre-processed to exclude punctuations, symbols, numbers, anonymous tags, foreign terms, web addresses, phone numbers, emojis, and email

ids. After this step, there were a total of 10,276, 495 tokens in the corpus. The number of distinct words in the corpus was W = 105,780 with the number of distinct lemmas L = 81,347. After stemming the lemmas, the number of distinct stems was S = 70,153. The ICF after lemmatization and stemming respectively was calculated as follows:

ICF1 = (W - L) / W = (105,780 - 81,347) / 105,780 = 0.231

ICF2 = (W - S) / W = (105,780 - 70,153) / 105,780 = 0.337

The fractional reduction in index size between lemmatization and stemming is:

ICF 3 = (L - S) / L = (81,347 - 70,153) / 81,347 = 0.138

In order to evaluate the stemmer qualitatively, Table 1 gives a list of the produced stems of the most frequent words in the CorCenCC corpus. The first column provides the 10 most frequently occurring words that do not change under the stemming rules, the second column gives the 10 most frequently occurring words which stem to words already present in the corpus, and the third column gives the 10 most common words that stem to word forms that were previously not present in the corpus. The stems' overall frequency-based rank is also given in parentheses. In highly morphological languages one would expect a large proportion of the stems to also be words themselves.

Table 1 Qualitative evaluation of stemmer output

No Change	To Words	To New
yn - yn (1)	'r - y (5)	cymru - cymr (32)
i - i (2)	'n - yn (7)	o'n - o yn (69)
y - y (3)	mae - bod (9)	gymraeg - gymra (70)
a - a (4)	yr - y (10)	i'w - i ei (73)
o - o (6)	ac - a (12)	cymraeg - cymra (75)
ar - ar (8)	oedd - bod (19)	dros - tros (80)
ei - ei (11)	sy - bod (26)	wedyn - wed (92)
wedi - wedi (13)	yw - bod (28)	achos - ach (112)
am - am (14)	'i - ei (29)	siarad - siar (124)
ni - ni (15)	fod - bod (30)	amser - ser (126)

The table shows that the most frequently occurring words either do not change or stem to known words, while only 7 of the top 100 words produce unseen words. Here, all words that do not change, the first column in table 1, correspond to words that do not need to change as they are in their simplest form. The second column highlights that some of the most frequently occurring words have irregular morphologies, with most being various forms of *bod*. These words were indeed stemmed correctly and sensibly. The final column shows the words that produce stems which are not words themselves. Many of these make sense, however, there are some instances, for example *achos* - *ach*, *siarad* - *siar* and *amser* - *ser*, where the stemmer has not produced the most obvious stem. All these cases are, however, infrequent, with none appearing in the top 100 words of our corpus. In addition, it is obvious

how the stemmer has mis-stemmed these. They all contain common suffixes or prefixes (*os*, *ad*, and *am*), which in most other cases would be desirable to strip from the word.

5 CONCLUSION

This chapter describes an adaptation of a Porter stemmer for the Welsh language. Stemming is one of the basic tools of linguistic pre-processing and as such is one of the key enablers for development of human language technology in a less-resourced language such as Welsh. The stemmer complements the existing activities in Welsh natural language processing. Currently, none of the existing tools supports stemming. For example, Welsh Natural Language Toolkit[4] implements a set of rule-based Welsh natural language processing tools for tokenisation, lemmatization and POS tagging. A set of tools with similar NLP capabilities have been implemented to support pre-processing of documents stored in CorCenCC, which are tokenised using CyTag, a rule-based POS tagger, which also performs lemmatization [17]. Another POS tagger, which can be used as a web service without the need to install it locally, can tag lexical categories (e.g. verbs and nouns), but also specifically Welsh features such as mutations [18]. The same team developed a lemmatizer, which can be used to normalise any inflected, mutated and/or conjugated word into its lemma [19]. Our Welsh stemmer can be combined with any of these tools as well as Welsh word embeddings [20] to improve semantic aspects of natural language processing in Welsh.

ACKNOWLEDGMENTS

This study was funded by the Welsh Government as part of a project in relation to Welsh language processing infrastructure.

REFERENCES

[1] M. F. Porter. 1980. An algorithm for suffix stripping. Program 14, 3 (1980), 130-137.

[2] Anjali G. Jivani. 2011. A comparative study of stemming algorithms. International Journal of Computer Applications in Technology 2, 6 (2011), 1930-1938.

[3] Richard W. Sproat and Maurice V. Wilkes. 1992. Morphology and computation. MIT press.

[4] Donna Harman. 1991. How effective is suffixing?. Journal of the american society for information science 42, 1 (January 1991), 7-15.

[5] Julie B. Lovins. 1968. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics 11, 1-2 (March and June 1968), 22-31.

[6] M. F. Porter. 2001. Snowball: A language for stemming algorithms. Retrieved May 16, 2021 from http://snowball.tartarus.org/texts/introduction.html

[7] M. F. Porter. 2006. Stemming algorithms for various European languages. Retrieved May 16, 2021 from http://snowball. tartarus. org/texts/stemmersoverview. html

[8] C. Paice. and R. Hooper. 2005. Lancaster stemmer. https://github.com/words/lancaster-stemmer

[9] Massimo Melucci and Nicola Orio. 2003. A novel method for stemmer generation based on hidden Markov models. In Proceedings of the twelfth international conference on Information and knowledge management. Association for Computing Machinery, New York, NY, United States, 131-138.

[10] Robert Krovetz. 2000. Viewing morphology as an inference process. Artificial intelligence 118, 1-2 (April 2000), 277-294.

[11] David A. Hull and Gregory Grefenstette. 1996. A detailed analysis of English stemming algorithms. Technical Report. Rank Xerox Research Centre.

[12] Jinxi Xu and W. Bruce Croft. 1998. Corpus-based stemming using cooccurrence of word variants. ACM Transactions on Information Systems (TOIS) 16, 1 (January 1998), 61-81.

[13] Fuchun Peng, Nawaaz Ahmed, Xin Li and Yumao Lu. 2007. Context sensitive stemming for web search. In Proceedings of the 30th annual

[4]https://sourceforge.net/projects/wnlt-project/

international ACM SIGIR conference on Research and development in information retrieval. Association for Computing Machinery, New York, NY, United States, 639-646.

[14] Steve Neale, Kevin Donnelly, Gareth Watkins and Dawn Knight. 2018. Leveraging Lexical Resources and Constraint Grammar for Rule-Based Part-of-Speech Tagging in Welsh. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). LREC, Miyazaki, Japan.

[15] William B. Frakes and Christopher J.Fox. 2003. Strength and similarity of affix removal stemming algorithms. In ACM SIGIR Forum (Vol. 37, No. 1). ACM, New York, NY, USA, 26-30.

[16] Dawn Knight, Steve Morris, Tess Fitzpatrick, Paul Rayson, Irena Spasić and Enlli-Mon Thomas. 2020. The National Corpus of Contemporary Welsh: Project Report | Y Corpws Cenedlaethol Cymraeg Cyfoes: Adroddiad y Prosiect. Cardiff University, Wales. arXiv:2010.055 42.

[17] Dawn Knight, Fernando Loizides, Steven Neale, Laurence Anthony and Irena Spasić. 2020. Developing computational infrastructure for the CorCenCC corpus: The National Corpus of Contemporary Welsh. Language Resources and Evaluation. https://doi.org/10.1007/s10579-020-09501-9

[18] Dewi Bryn Jones, Patrick Robertson and Gruffydd Prys. 2015. Welsh language Parts-of-Speech Tagger API Service. Bangor University, Bangor, Wales. http://techiaith.cymru/api/parts-of-speech-tagger- api/?lang=en

[19] Dewi Bryn Jones, Patrick Robertson and Gruffydd Prys. 2015. Welsh language lemmatizer API service. Bangor University, Bangor, Wales. http://techiaith.cymru/api/lemmatizer/?lang=en

[20] Geraint Palmer, Padraig Corcoran, Laura Arman, Dawn Knight and Irena Spasic. 2021. A closer look at Welsh word embeddings. In Language and Technology in Wales: Volume 1. Bangor University, Bangor, Wales.

Developing ColloCaid: a visualization and text-editing tool to help writers with collocations

JONATHAN C. ROBERTS

School of Computer Science and Electronic Engineering, Bangor University, Wales

PETER W. S. BUTCHER

School of Computer Science and Electronic Engineering, Bangor University, Wales

ROBERT LEW

Faculty of English, Adam Mickiewicz University, Poznań , Poland

GERAINT REES

Centre for Translation Studies, School of Literature and Languages, University of Surrey, England

NIRWAN SHARMA

Faculty of Science, Technology, Engineering & Mathematics, Open University, England

ANA FRANKENBERG-GARCIA

Centre for Translation Studies, School of Literature and Languages, University of Surrey, England

We have developed the ColloCaid text editor, a writing assistant that suggests natural collocations for words as the users type them. The tool also visualizes words to clarify word associations and presents alternative words. Developing a visualization tool and editor involves many choices, challenges, and decisions during the design, development, and evaluation of the tool. Being a collaborative project, ColloCaid has brought together researchers with different skills, backgrounds, and work styles. In this chapter, we discuss our development cycle when creating the ColloCaid tool, from the selection of the reference text corpus data to designing and developing appropriate word visualizations. Our account can be of interest to other researchers, offering insights on the development of interactive linguistic tools that incorporate visualizations.

Keywords and Phrases: visualization, linguistics, corpus linguistics, close writing

1 INTRODUCTION

When writing academic texts, choosing the right words can be a difficult challenge. In this work we focus on helping writers to choose the best words, especially facilitating access to typical word partners: strong collocations. With this purpose in mind, we have developed the ColloCaid text editor and word visualization tool using a lexical database built from quality text corpora to offer writers a selection of collocations as they work on their text.

Approaching a new topic can be challenging to writers. Authors who have been academics for a long time tend to be familiar with the words and phrases commonly employed in the academic domain. Their written text flows well, has a structure that is appropriate for the domain and the key words they use are relevant for the topic. By

contrast, novices may find academic writing a daunting task, as may writers who are not writing in their first language, because they do not necessarily know the words or phrases typical of that domain. They may choose to use a word that has a similar meaning, but is not conventionally used in that context, or use words that do not usually go well together, producing an inappropriate lexical collocation. Krishnamurthy [1] defines collocation as words that occur "with a greater frequency than the law of averages would lead you to expect". The phrases appear because of repeated use in a given context. Knowledge of collocations is important to present natural (and so clear) sentences. Incorrect collocational choices result in awkward prose, or stand out as being unusual, thus distracting readers from the content.

Collocations occur in all domains, though they are not necessarily the same collocations. For example, someone writing about computers may say that they have a powerful computer – a computer that achieves many petaflops of computing power and has many computing cores. While in general other adjectives might potentially be used as a replacement for powerful, such as muscular, sturdy, or robust, as found in a thesaurus, they would either be inappropriate or express a different meaning. Thus, a *muscular computer* would be a very unusual and cryptic word combination, while a *sturdy computer* would mean something different: a computer capable of withstanding physical stress, though not necessarily one that performs operations fast. In the field of visualization, authors use terms such as *bar chart* in preference to *bar plot*. Why is this the case? Certainly, the words *chart* and *plot* could be interchanged, indeed authors prefer *pie chart* over *pie plot* [2]. They have very similar meanings, and each version could be appropriate. But other authors in the field have typically chosen one way, over another, to express their ideas. The more often we come across a word combination, the more it becomes familiar, entrenched, and thus natural, becoming a strong collocation. Weak collocations are the opposite: they are less common, sound unnatural and make the text awkward to read.

Collocations can be analysed by examining large digital collections of existing texts serving as evidence of language use. The analysis process loads machine-readable texts and, using statistical models, provides the user with useful information about the word. In this situation, the input data (the machine-readable texts) and the output data (statistical and distributional properties of the words) help the writer to understand the domain. Subsequently, a writer can query a word to find its conventional collocation, as well as draw on the original texts to explore example sentences so as to help them select appropriate terms; "corpus linguistics is not concerned with what is possible in a language, but in [sic] what is probable'' [3]. Thanks to corpora, writers can choose the words they need to communicate their ideas effectively [3], [4]. Corpus query software tends to be powerful enough to deliver comprehensive information about the analysed corpus, yet is too difficult to use for non-linguists. Most users will not be aware of how the corpora can be useful to them, or how to build relevant queries to help them in the writing. It is also easy for writers to get distracted and follow irrelevant words, and even potentially they can misinterpret the meaning of the results. What is required is a way to integrate the benefits of corpus linguistic tools with an editor.

By developing a specific text editor that incorporates linguistic and corpus analysis, a writer would look up specific collocations and see enough alternatives to make an informed choice, yet not as many so as to get confused. Appropriate visualization can also help in effectively presenting the choice of alternative words. With this tool, users could examine the collocations in-situ. This has the advantage that there is no need to jump away from the editor, thus any potential disruption to the user's writing flow is avoided. The tool also provides access to methods that encourage users to investigate and explore new words and to improve their writing skills. In this chapter we present

the ColloCaid tool and explain how it was developed. We put the work in context in Background and Goals, then discuss its design, development, and evaluation.

2 BACKGROUND AND GOALS

Developing any linguistic tool requires a broad set of researchers with different skills. The ColloCaid project [5, 6, 7] brings together researchers with skills in lexicography, language teaching, pedagogy, human computer interaction, visualization and computer programming. The project has funding for four years, from the Arts and Humanities Research Council (AHRC) and is a collaboration between academics in Surrey, Bangor (UK) and Poznań, Poland. The overarching aim of the project is to design and develop an interactive editor that could be used by Academic English writers to learn and use collocations. The editor allows writers to type words, and the system highlights and lists collocations. The underpinning data has been carefully and deliberately curated by the team, and the human interface and visualization front-end has gone through many revisions and user-evaluations, to assess its usability. Readers can try out the tool and explore different collocations by visiting the collocaid.uk website. ColloCaid analyses the words that people write, recommends and visualizes different words based on systematic data from specialized text corpora.

With advances in corpus linguistics, and development of online tools, it is now much easier for any user to access and query existing corpora. In fact, many of the online corpus linguistic tools offer users easy access to pre-analysed corpora, such as British National Corpus (BNC) and the Corpus of Contemporary American English (COCA). Some also allow users to upload their own documents for analysis. Tools such as AntConc [8], CQPweb [9], SketchEngine [10] and Wmatrix [11] allow the more ambitious users to create their own custom corpora.

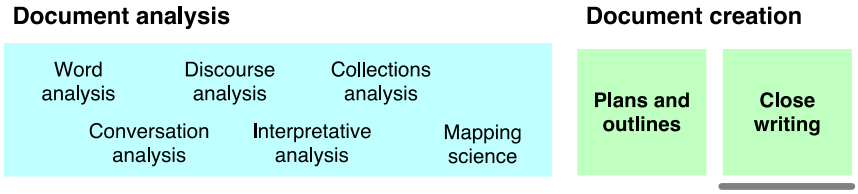

Figure 1: Corpus linguistics can help writers analyze different structures in written texts or text as it is created. With the ColloCaid project our focus is to help authors as they write and compose their texts.

There are many reasons to analyse text, we classify them into two distinct groups: see Figure 1. First, scholars may wish to analyse text documents that have previously been written by other people. They may do this to understand how to write similar documents, or texts in the style of another author, or they may wish to understand the structure of many documents and understand how different versions of the information have changed. Second, people may wish to analyse the document that they are writing. Dictionaries, thesauri, grammar checkers and so on help writers to choose appropriate words and make their documents easy to understand.

There has been a huge breadth of research carried out in the area of document analysis. Scholars have applied corpus linguistics techniques to understand the structure of sentences and paragraphs such to analyse sentiment or grammar, analyse how writers have organized the complete document, or how many collections or versions fit together, or even map how science has adapted (Figure 1). Tools enable researchers to investigate different aspects of the documents, from looking at minute detail of sentence structures to overarching structure. This analysis can

help researchers discover how authors wrote their documents (through document analysis), which can help writers to write better, as they are learning from other authors.

The focus of our work is to help writers create documents. We use the term *close writing* as a counterpart to *close reading*. Close reading is a technique that many students know. They are asked to analyse a passage, take notes, understand narrative voice, tone and critically contemplate the kinds of words and sentences that a particular author has used. Similarly, with close writing authors must think carefully about the words they use, put themselves in the shoes of their reader and reflect how their work will be viewed and understood. In particular, choosing appropriate collocations will help them write better and more understandable texts. Tools to help people write better utilize linguistic techniques that are used in document analysis. Lexicology, grammar, discourse analysis, stylistics, and so on, can help inform authors to choose the best words and phrases. Data (as words) are analysed and the results are displayed as information to inform writers. It is a pattern-based approach, where writers see good examples and can apply the knowledge to their own context. This is data-driven learning [12], [13], [14], [15], [16].

Today, there are many tools that help people write texts. Word processors, such as Word™, Apple Pages or LibreOffice allow users to write and edit texts. Online editors, such as Microsoft's Word online or Google docs help writers share documents and access their texts remotely. Specialist editors help with particular challenges, for example OneNote or Evernote allow users to create, edit and share notes. Most word processors today integrate editing and formatting functionality – most employ WYSIWYG formatting [17] (what you see is what you get) to enable the author to view the final version. Some processors separate editing from formatting; plain text editors allow users to write ASCII text, which can be formatted using markup languages such as LaTeX, HTML or markdown. Most text editors show spelling errors and make suggestions and improvements to grammar. Text editors have improved over the years, from having limited functionality to integrating many writing aids. For instance, speech-to-text and text-to-speech functionality is readily available in most systems. Grammar checking has been improved, for example by integrating systems such as Grammarly.

However, there is much research to be done and many techniques that could be integrated better with text editors. For instance, there are many systems that calculate the readability of text, to profile written text against vocabulary lists such as Academic Word List (AWL) [18] or New Academic Word List (NAWL) [19], but these systems are separate to the editor. In most systems, the writer needs to copy and paste their text into a separate analysis system to get results. Many systems visualize text documents to display relationships between words or documents, but again are not integrated well with text editors. Finally, there are many tools that create bespoke text corpora of document bodies, though again these are typically separate to the editor. Having the information in a separate tool means that it is not necessarily quick or easy to use. It can be daunting for learners to navigate and explore the information that is presented to them and difficult for them to choose the best words. In other words, it is often difficult for writers to critically write using the tools offered to them. Our focus is to better integrate corpus linguistics, collocation analysis and word visualization with a text editor, to enable writers to make better word choices, as well as to learn, so as to become better writers.

3 DEVELOPMENT METHODOLOGY

When designing and developing any tool, there are many questions to ask and decision to make [20]. These include: how long is the development or what resources are available? What is the purpose and job of the tool? What data does the tool operate on? Understanding the purpose of the tool is important. Without a clear vision of where the

tool fits in with other tools, not realising the resources available, or knowing how it could be used, would make the project difficult to develop. We knew at the start of the project what we wanted to create: a tool to allow writers to edit the documents and recommend collocations. We knew also about the data, and would create, using specialized corpora, a database of collocations with some 30,000 items. While at the start we had a holistic vision of our completed tool, we did not know the minute detail or specific functionality of the tool. Developers need to be able to create a tool that will work and be fit for purpose. In the news and on social media there are many examples of computing projects that do not complete and those that are unusable. Subsequently, careful planning needs to be performed to make sure that the project will deliver on time. Especially with many research projects it is difficult to exactly describe the full functionality of the tool at the outset. Because ideas change and new solutions appear as the project continues, it is difficult to create a fully defined specification at the start of a project. Consequently, we believed that it was impossible to create a full specification at the start of the project, and through discussion decided to use an Agile design and development approach [21].

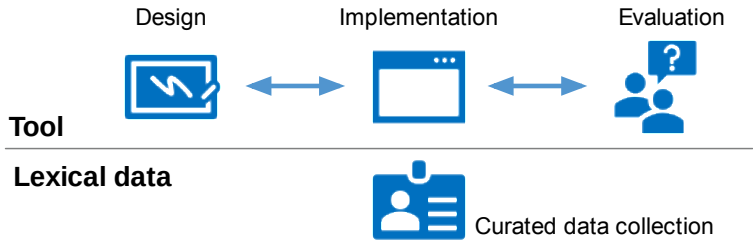

Figure 2: In the project we started implementing early, to create an early (low functional) working prototype. We refined the implementation through iterating many prototypes. At each stage we tested the tool internally, and evaluated the system with student users. Subsequently we improved the design and implementation each evaluation cycle.

We created several prototypes throughout the project and evaluated them as the project progressed, as shown in Figure 2. We divided the tool into two parts: the underlying lexical data, and the human interface tool. We also followed other Agile principles. One important Agile principle we followed was to always keep a working version of the software. This was important, because we wanted to have a version that we could demonstrate at short notice. In fact, throughout development we had several parallel versions. First, a public working version, with functionality that we were willing to make public. Second, a non-public version with improved functionality, to test on users and which helped us explore different solutions. Third an in-development version with code that may be only partially complete. These principles are explained in the Agile Manifesto [22], including: developing tools to satisfy customers through early and continuous delivery of the software; welcome changing requirements even late in the project; delivering working software frequently; building software around motivated individuals; keeping the output simple; and reflecting – as a team – on the output and deciding how to improve it.

Reflection was another major Agile principle we followed. As a team we regularly reflected on the tool. We held regular meetings as a team, discussing the output and the current version of the tool. We tested the tool ourselves and noted any errors. We only had a small development team, with two post-doctoral researchers' effort, and because we kept working versions of the software, syntactical and major errors that stopped the tool from working were solved outside these meetings, leaving the meetings to discuss semantic errors. One postdoc focused on

linguistic and database aspects of the project, with the other postdoc focusing on developing the interface (see Figure 2). Other members helped with the design, evaluation, dissemination, presentations, and so on. The team would meet regularly to discuss the current implementation, its functionality, errors, and improvements. Furthermore, we evaluated the tool with real users. We were fortunate to have access to many student learners across different countries and of varying abilities, from beginner undergraduate students to students on advanced master's linguistic programmes. We evaluated the ColloCaid tool on hundreds of users over the project duration.

4 COLLOCAID DESIGN – UNDERPINNING DATA AND USER INTERFACE

We followed a further important design principle: one of simplicity. While we could provide a tool that would deliver hundreds of possible collocations, it could confuse the user: balance needs to be met between functionality, ease of use, and comprehensiveness of word coverage. It would not be good for a writer to see all possible words and not be able to use the tool. If too many words are displayed, users would get confused as to which one is important and be unclear what words to choose. Instead, the tool needs to be carefully designed to present a suitable set of candidate words – to give the writer a choice of words – but be easy and clear to use.

Our solution was to develop an underpinning collocational database which was carefully curated from frequent words that have been discovered by scholars in British academic writing. By focusing on the most frequent words, we were able to develop a suitable tool to help writers of academic English [6]. Our underpinning database consists of over 30 thousand co-occurring words [7]. We derived the database by carefully considering lists of common words by other researchers, including words commonly found in student writing across different disciplines [23], and academic keyword lists [5, 24]. Figure 3 summarises the underpinning database, which is based on vocabulary lists, quantities, and examples from corpora of professionally published academic writing, and uses automatic lemmatisation. Table 1 shows an example of our collocation data; with the base word we store the collocate, along with the frequency and association score. An indicative example is shown in the table.

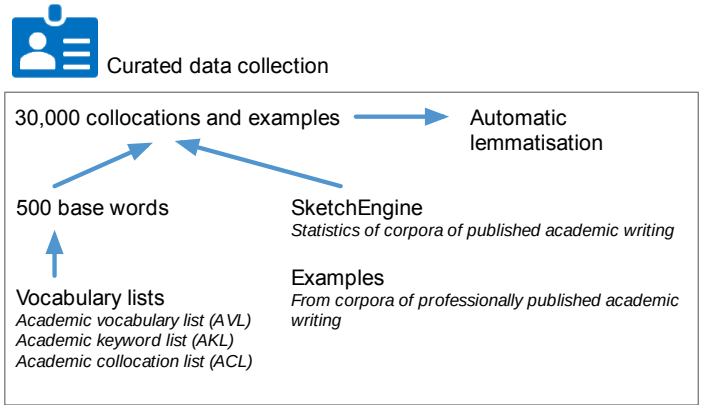

Figure 3: ColloCaid data is created from SketchEngine [10], incorporates many example collocations, and is underpinned from over 500 base words derived from academic vocabulary, collocation, and keyword lists.

Table 1: ColloCaid example data, showing base with collocation, different data values and an example.

Base	POS	Collocate POS	Collocate	Freq	Score	Example
access	n	v	gain	534	10.46	unauthorized users can gain access to it

Base	POS	Collocate POS	Collocate	Freq	Score	Example
access	n	v	ensure	113	8.09	this could also ensure access to information
access	n	v	restrict	114	8.66	the state has restricted access by foreigners
access	n	v	grant	73	8.05	the plan appeared to grant sole access to media for joint activities
access	n	v	secure	79	8.19	the first is securing universal access to justice
access	n	v	facilitate	114	8.33	technologies that might facilitate open access to these resources should be a priority
access	n	v	have	1818	8.49	populations who have access to online research
access	n	v	improve	139	8.09	improving access to prevent delays in answering
example	n	v	illustrates	147	10.56	this example also illustrates a special case of the two constraints
example	n	v	concerns	35	8.11	the next example concerns different interpretations
example	n	v	relates	13	6.78	the most relevant example relates to the manipulation of party primaries
example	n	v	inspires	13	7.59	many crimes have supposedly been inspired by examples shown on film
example	n	v	demonstrates	26	7.61	its nature can easily be demonstrated by an example
example	n	v	includes	504	10.85	examples include provocation and diminished responsibility
example	n	v	abounds	12	7.92	examples abound of some uses of the concept of gender
example	n	v	shows	110	8	an interesting example shows a group partition

Simplicity was also an important strategy that we wanted to follow when designing the user interface. In fact, many design researchers have created user-interface design guidelines that include a recommendation to keep things simple. For instance, number eight of Jakob Nileson's 10 heuristics for design is to aim for an "aesthetic and minimalist design" [25]. The idea of keeping things simple, along with our wish to design solutions that are usable, fits well with the Agile strategy. However, there are many different design strategies that could be followed [26], and it can be difficult to know which to choose. On the one hand, with the Agile approach we wanted to start developing the tool, while on the other hand we wanted to produce a separate critical design study. Within our Agile development, we used all these strategies.

First, we created an early prototype of ColloCaid using PowerPoint™. This is a *pretotype* [27] a low fidelity mock-up that we used to demonstrate the principal ideas. This pretotype version helped us promote our research at an early stage. We used it in presentations and showed it to interested researchers. This gave us a good idea of the reaction of people. In fact, many people gave us feedback saying that they would use it. Their reactions inspired us and helped us to refine our ideas.

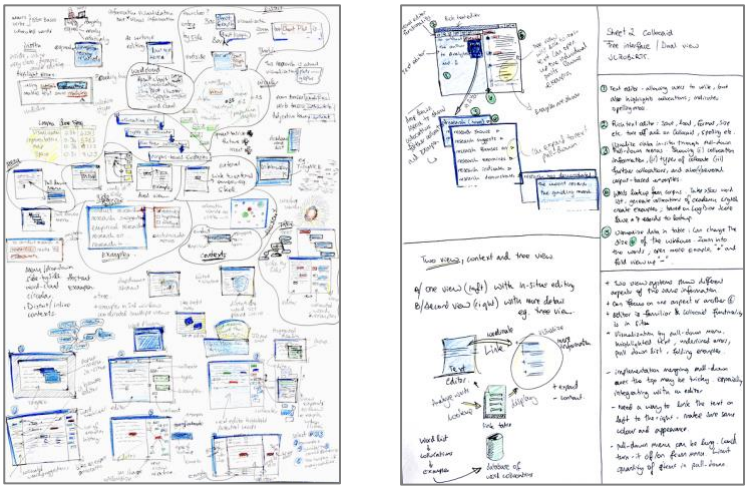

Figure 4: Scans of designs showing the first and second sheet from the Five Design-Sheet process. Left shows a page of many different ideas, while the right page demonstrates one design of the final tool that shows a two-view system, with the text editor on the left and the visualization on the right.

Second, we sketched different design solutions. We wanted to investigate different alternative ideas and explore potential interface solutions. Sketching is useful, because it is cheap, relatively quick to create and can be done at any time. It is quicker to sketch different solutions than it is to develop and implement it in code. While ad hoc sketching [28] is popular, we chose to use the Five Design-Sheet (FdS) sketching methodology for interface design [20], [29], [30]. We chose the FdS methodology because we have used it successfully with other projects, and it provides a structured workflow from contemplating many different design ideas, to three more complete versions, and a final realisation version. We used A3 sheets of paper, and chose a reduced colour pallet of blues, red and yellow for highlight. Figure 4 shows two sheets from this process. Sheet 1 demonstrates many different design ideas, whereas the second page expands on one particular solution, focusing on the idea of a linked dual view system: with the text editor on the left and the visualizations shown on the right. When a user clicks on something in the text editor, it can be displayed in more detail on the right-hand side.

5 TOOL DEVELOPMENT AND IMPLEMENTATION

When developing any tool that relies on a text editor, one of the main choices to make is whether to write code from basic principles, develop an add-on for a current editor, or adapt code from open-source software. There are advantages and disadvantages for each of these solutions, and the decision needs careful discussion within the development team because it changes how the tool will be used. We spent many long hours deliberating each method.

Building the editor from scratch would give us full control over the code. We would be able to adapt it to our needs, it would be easy to extend it at a later stage and we could design it with functionality specific to the project. We would also have full rights over the code and could integrate it into any project we wished. However, this is a challenging solution that requires a lot of engineering. It requires the programmer to understand parsing, word and character manipulation, text editing, user-interface code development and so on. It can be challenging for a developer to create code that works appropriately, without errors, and it would certainly take much time and effort

to code. In addition, one of our desires was to allow writers to be able to copy and paste code from other projects and have functionality that is readily found in a commercial editor, such as font changes, headings, colour editing. Because of the challenges with coding, we chose not to follow this solution.

Another potential solution would be to create an add-on. There are several possible ways that this could be achieved. It could be an extension to a browser, or an extension to a word processor such as Microsoft Word. In fact, Microsoft Office have provided different ways to extend its functionality. But when we started the ColloCaid project, we found it extremely difficult to find information and examples of how to extend Word. Documentation on how to extend Word was not easily accessible, much specialist knowledge was required, and it was challenging to write effective programs. Consequently, we took the decision not to extend Word. In particular we were unclear how to integrate a database, highlight individual words, integrate with the spell checker, and highlight and change words. This was before the recent improvements in Office add-in technology. The new add-in functionality now allows developers to adapt Microsoft Word, programmers can use the Word JavaScript API and the Office JavaScript API, to incorporate different databases, perform single sign-on (SSO) security, use Ember, Backbone, Angular and React. The idea is that code that could be run in a browser can be run as an Add-in to Word, and programmers can create requests to act on Word objects and synchronize object state.

The third solution is to adapt an open-source editor. There are many editors that we could select. Desktop applications such as Atom have a community of developers, GitHub packages setup and cross platform support. Similarly, the brackets editor offers cross platform support and GitHub community. An alternative is to develop online. Software such as Quill, EditorJS and TinyMCE are HTML editors that provide an online WYSIWYG environment. After deliberation, we decided to develop an online tool, and opted for the TinyMCE editor, which uses GNU Lesser General Public License. However, our decision to use TinyMCE still presented challenges: in order to extend it properly, we needed to understand how the TinyMCE works, and how the code was written. While overcoming these challenges did take much time, we managed to make several additions to the code, and once a new feature was added, we rigorously tested it.

We extended TinyMCE to provide the functionality that we required. Figure 5 shows a screenshot of the editor. We parse the words that have been written, looking them up in our collocation database. Words that are in the database are shown to the writer by a green dashed underline. We added in-situ drop-down menus, which appear when a writer clicks on an underlined word. The drop-down menu allows writers to select different parts of speech, specific collocations and examples. Writers can click on any of the words in the menu to insert them into the document.

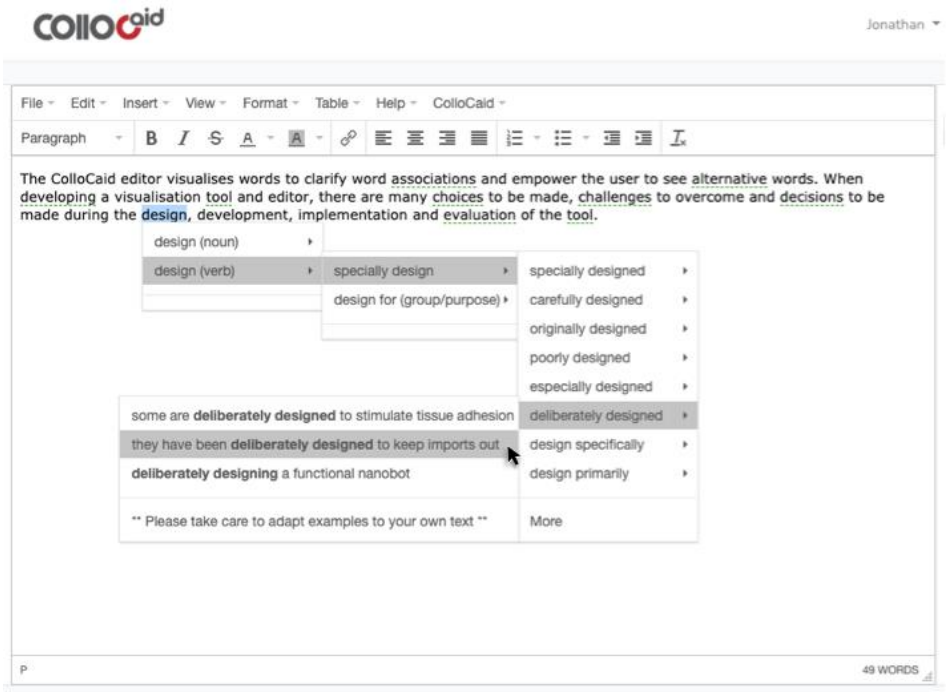

Figure 5: The ColloCaid editor showing the drop-down menus. Writers can make choices over the data they see, and any element they select gets inserted into the text.

We have designed several alternative visualizations. We designed tables, lists, circle visualizations and other styles. We have implemented some of these ideas. Figure 6 shows our Tree and Fan Views. With the Tree View (left-hand panel) users can select the triangle icons to unfold detail about the collocations and show example sentences. As with all our visualizations, when the user clicks on a line, the text is inserted into the document. The right-hand panel in Figure 6 shows a screenshot of the Fan View, which demonstrates what collocations can be used before, or after, the selected word. In all visualizations, when the writer selects "Show More", a list of additional words is included. When the list is longer than the screen, a scrollbar is added. In addition, more information can be displayed for each of these words, and we provide a direct link to a list exemplifying the word pattern selected in SkELL (Sketch Engine for Language Learning). SkELL is a tool to help people explore how specific terms are used by real speakers of English, and provides good examples of collocations and synonyms identified automatically from a large corpus.

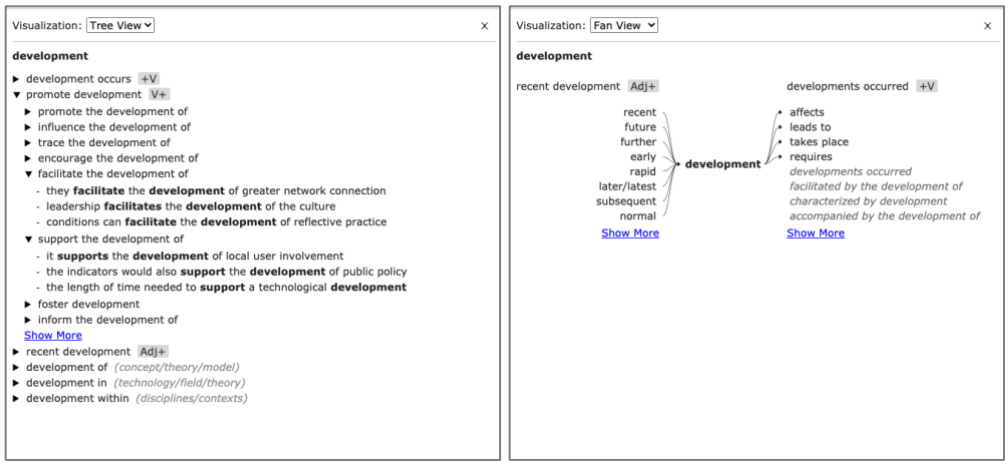

Figure 6: Two visualizations in the ColloCaid tool. The panel on the left shows the Tree View, where users can click on the triangles to open or close additional information. The panel on the right shows a screenshot from the Fan View visualization on the word development. This depicts collocated words that could be placed before, or after, the selected term. Lists of additional words are depicted when the writer selects 'show more'.

6 EVALUATION AND FEEDBACK

There are many ways to evaluate tools [31]. We needed to follow a simple, and quick to perform, evaluation strategy that would fit in with the Agile development methodology. We also wanted a method that we could use with students, that would be easily understandable in different languages. We could ask questions such as whether the writer would use the tool again? We could ask questions about efficiency, learnability, helpfulness, attractiveness and so on, and create our own bespoke set of questions. However, we decided to follow the System Usability Scale (SUS) [32]. The SUS consists of ten simple questions, which participants answer on a 5-point Likert scale, from "strongly disagree" to "strongly agree".

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Using this method, we can gain quantitative data on the perceived usability of the system. In addition, we decided to add two further open-ended questions, asking participants to explain something they liked, and something else they disliked, about the tool. We used the SUS because the ten questions have been translated into many languages (and so we were able to readily use the translations, such as with our Polish testers), we have used it in other

projects, and the ten questions can be quickly administered to the participants without the need for long explanations.

We used this questionnaire to evaluate the tool at different stages of development. We evaluated the tool over 140 participants across five sites: León, Paris, Porto Alegre, São José do Rio Preto, and Poznań, and received much feedback from users. For example, one of our early prototypes had a table at the bottom of the editor window that showed a complete list of collocations from the current text. The table would update as the writer progressed. However, through the evaluation users said that the text "danced" around, and that the position of the table (below the editor) was not useful. Responding to this feedback, we removed the Table View, and instead implemented the Tree View (see Figure 6).

7 SUMMARY AND CONCLUSION

We used an Agile development methodology and continuously develop the tool from early prototypes to the completed version and have evaluated ColloCaid with real users. We summarize five aspects that have helped us succeed. First, we spent much time and effort to curate the underpinning collocation data. It is important to create trustworthy data, without collocation data and real examples the tool would not be useful. Second, one of the important aspects of our Agile approach was to keep two working versions of the tool and one in-development version. This meant that we were able to demonstrate and present our work at any time during the project. Third, by extending the TinyMCE editor we were able to develop a tool that operates in a traditional way, with writers operating a familiar interface. Fourth, the design, development and evaluation of different visualization ideas, and our emphasis on simplicity, has created useful visualizations. Finally, our programme of ongoing testing and evaluation, integrated with the System Usability Scale (SUS) enabled us to engender useful and important feedback on the tool that has helped with ongoing improvement.

In conclusion, we have presented the ColloCaid editor, which is a writing tool that assists writers as they edit texts. The tool suggests natural collocations for words when the user types them and visualizes words to provide alternative suggestions.

ACKNOWLEDGMENTS

This work is funded by Arts and Humanities Research Council (AHRC) grant AH/P003508/1, http://collocaid.uk

REFERENCES

[1] Ramesh Krishnamurthy. 1987. The process of compilation. In Looking up: An account of the COBUILD project in lexical computing. Collins ELT, London and Glasgow, 62-85.

[2] Jonathan C. Roberts, H. Al-Maneea, Peter Butcher, Robert Lew, Geraint Rees, Nirwan Sharma and Ana Frankenberg-Garcia. 2019. Multiple Views: different meanings and collocated words. Computer Graphics Forum 38, 3 (June 2019), 79-93. 10.1111/cgf.13673

[3] Alex Boulton. 2016. Integrating corpus tools and techniques in ESP courses. Asp 69, 69 (March 2016), 113-137. 10.4000/asp.4826

[4] Ana Frankenberg-Garcia. 2018. Investigating the collocations available to EAP writers. Journal of English for Academic Purposes 35 (July 2018), 93-104.

[5] Kirsten Ackermann and Yu-Hua Chen. 2013. Developing the Academic Collocation List (ACL)--A corpus-driven and expert-judged approach. Journal of English for Academic Purposes 12, 4 (December 2014), 235-247.

[6] Ana Frankenberg-Garcia, Robert Lew, Jonathan C. Roberts, Geraint Paul Rees and Nirwan Sharma. 2019. Developing a writing assistant to help EAP writers with collocations in real time. ReCALL 31, 1 (January 2019), 23-39.

[7] Jonathan C.Roberts, Peter W. S. Butcher, Robert Lew, Geraint Paul Rees, Nirwan Sharma and Ana Frankenberg-Garcia. 2020. Visualising Collocation for Close Writing. In EuroVis 2020 - Short Papers. The Eurographics Association, Online.

[8] Laurence Anthony. 2018. Visualisation in corpus-based discourse studies. In Corpus Approaches To Discourse. Routledge. London. 197-224.

[9] Andrew Hardie. 2012. CQPweb - combining power, flexibility and usability in a corpus analysis tool. International Journal of Corpus Linguistics 17, 3 (January 2012), 380–409.

[10] Adam Kilgarriff, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlý and Vít Suchomel. 2014. The Sketch Engine: ten years on. Lexicography 1, (July 2014), 7-36.

[11] Paul Rayson. 2008. From key words to key semantic domains. International Journal of Corpus Linguistics 13, 4 (January 2008), 519-549.

[12] Tim Johns. 1991. Should you be persuaded: Two samples of data-driven learning materials. English Language Research Journal 4, (1991), 1-16.

[13] Ana Frankenberg-Garcia. 2012. Raising teachers' awareness of corpora. Language Teaching 45, 4 (October 2012), 475-489.

[14] Ana Frankenberg-Garcia. 2020. Combining user needs, lexicographic data and digital writing environments. Language Teaching 53, 1 (January 2020), 29–43.

[15] Ana Frankenberg-Garcia. 2016. Corpora in ELT. In The Routledge Handbook of English Language Teaching. Routledge. London. 401-416. 10.4324/9781315676203

[16] Alex Boulton and Tom Cobb. 2017. Corpus Use in Language Learning: A Meta-Analysis. Language Learning 67, 2 (December 2017), 348-393.

[17] Brad Myers. 1998. A brief history of human-computer interaction technology. Interactions 5, 2 (March/April 1998), 44-54. 10.1145/274430.274436,2,44-54

[18] Dee Gardner and Mark Davies. 2014. A new academic vocabulary list. Applied linguistics 35, 3 (July 2014), 305-327.

[19] Averil Coxhead. 2000. A New Academic Word List. TESOL Quarterly 34, 2 (Summer 2000), 213-238. https://doi.org/10.2307/3587951

[20] Jonathan C. Roberts, Chris Headleand and Panagiotis D. Ritsos. 2016. Sketching Designs Using the Five Design-Sheet Methodology. Transactions on Visualisation and Computer Graphics 22, 1 (January 2016), 419-428.

[21] Barry Boehm. 2002. Get ready for agile methods, with care. Computer 35, 1 (January 2002), 64-69. 10.1109/2.976920

[22] Martin Fowler and Jim Highsmith. 2001. The agile manifesto. Software Development 9, 8 (2001), 28-35.

[23] Sian Alsop and Hilary Nesi. 2009. Issues in the development of the British Academic Written English (BAWE) corpus. Corpora 4, 1 (May 2009), 71-83.

[24] Magali Paquot. 2010. Academic vocabulary in learner writing: From extraction to analysis. Continuum. London, New York.

[25] Jakob Nielson. 1994, 2020. 10 Usability Heuristics for User Interface Design. https://www.nngroup.com/articles/ten-usability-heuristics/

[26] Alex Bigelow, Steven Drucker, Danyel Fisher and Miriah Meyer. 2014. Reflections on How Designers Design with Data. In Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces. Association for Computing Machinery, New York, NY, United States, 17-24.

[27] Marc Rettig. 1994. Prototyping for Tiny Fingers. Communications of the ACM 37, 4 (April 1994), 21–27.

[28] Bill Buxton. 2010. Sketching user experiences: getting the design right and the right design. Morgan Kaufmann.

[29] Jonathan C.Roberts, Christopher J. Headleand and Panagiotis D. Ritsos. 2017. Five Design-Sheets -- Creative design and sketching in Computing and Visualization. Springer International Publishing. Switzerland.

[30] Jonathan C.Roberts. 2011. The Five Design-Sheet (FdS) approach for Sketching Information Visualization Designs. In Proceedings of Eurographics 2011 - Education Papers. The Eurographics Association, Llandudno, Wales, 27-41.

[31] Jeff Sauro. 2015. SUPR-Q: a comprehensive measure of the quality of the website user experience. Journal of Usability Studies 10, 2 (February 2015), pp.68–86. 10.5555/2817315.2817317

[32] Aaron Bangor, Philip T. Kortum and James T. Miller. 2008. An empirical evaluation of the system usability scale. International Journal of Human Computer Interaction 24, 6 (July 2008), 574-594.

Modelling and Processing Welsh Text using the TAWA Toolkit

LEENA SARAH FARHAT

School of Computer Science and Electronic Engineering, Bangor University, Wales

WILLIAM JOHN TEAHAN

School of Computer Science and Electronic Engineering, Bangor University, Wales

This chapter provides a discussion of the preliminary results of applying the compression-based TAWA NLP toolkit to various tasks in Welsh NLP using the CEG, CorCenCC and UAGT-PNAW corpora.

Keywords and Phrases: TAWA Toolkit, compression-based language modelling, CEG, CorCenCC, UAGT-PNAW

1 BACKGROUND AND MOTIVATION

This chapter describes a preliminary investigation into applying the TAWA toolkit [1] to the problem of modelling and processing Welsh text.

The Text Analysis from Waikato toolkit, known as TAWA, is a toolkit that applies compression-based language modeling for text analysis, text mining and NLP. This toolkit has been in continuous development for over 20 years. Originally implemented in C, it now has a range of user-friendly tools as well as more powerful libraries being developed in Python.

The toolkit has been successfully applied to a wide-range of tasks in text analysis, text mining and NLP often producing state-of-the-art results in many of these applications [1]. Some sample applications include: English-Chinese and English-Arabic bilingual sentence alignment; cryptanalysis of transposition ciphers; classification and segmentation of Arabic text; emotion recognition in English and Arabic text; gender and authorship categorization of Arabic tweets; and identification of gene function in biological publications.

A question arises as to how successful the toolkit would be when applied to the Welsh language. The main motivation behind this chapter is to produce preliminary work to investigate this question.

1.1 Compression-based Language Modelling

The TAWA Toolkit currently uses the character-based Prediction by Partial Matching (PPM) compression algorithm [2] which has been at the cutting edge of compression algorithms for over three decades and remains one of the best performing lossless compression programs for natural language text. The use of compression for Natural Language Processing (NLP) is of interest because at its core, compression gets rid of redundancy from data and any model created can be verified. Any prediction a model makes can be checked by encoding it then decoding it. TAWA is able to classify text using Minimum Cross-Entropy. This uses compression *codelength* (the number of bits required to encode a text string), seen as the cross entropy in this case, as a way of measuring the "goodness" of the created language model.

Figure 1 provides a visual representation of how this works. The diagram shows two models that have been created by training on text representative of two languages – English and Welsh – using the training tool provided by the toolkit. These models are then used by the classification tool to compress a specific testing text, whose uncompressed size is 1410 characters, as shown in the figure. The tool determines that the Welsh model compresses the testing text best, requiring just 2824 bits or 2.00 bits per character ("bpc"). This is a significant

compression rate, and a common result with natural language texts. It is also significantly better than the English model at compressing the text (5256 bits or 3.73 bpc). Again, this difference in compression between the English model and Welsh model is typical. This strongly indicates that the language in the testing text is most likely Welsh.

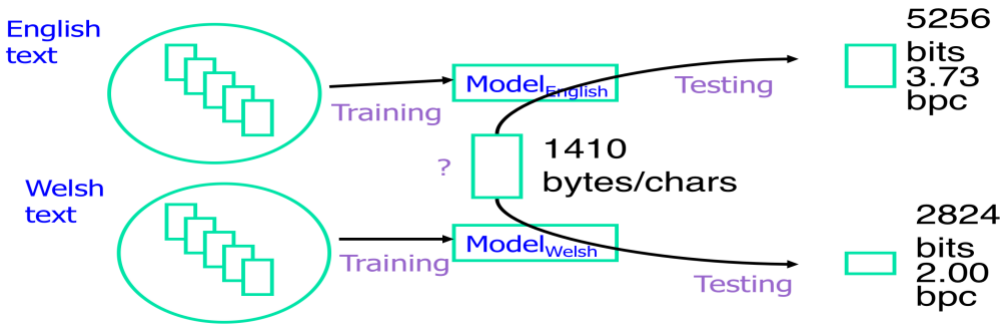

Figure 1: Visual representation of how compression is applied in TAWA for the language identification task. The smaller compressed output size for the Welsh model on the testing test indicates that the language is most likely Welsh.

TAWA also implements a noiseless channel model architecture for correction-based applications, by recasting them as a type of tag insertion problem [1]. All of this is brought together within the toolkit by making easy-to-use correction-based tools available for users to apply to various NLP tasks such as segmentation and tag insertion (markup).

The range of applications provided within the toolkit are shown in table 1.

Table 1: The range of applications provided within the TAWA toolkit

Application	Description
align	A compression-based application for verifying sentence alignment in a parallel corpus.
classify	A text classification tool for finding the training model that compresses the testing text best.
codelength	A tool for computing compression codelength without coding which can be used to predict the efficiency of a given compression.
decode	A tool that decompresses the compressed input file from a physical file on disk.
encode	A tool that compresses an input file, writing out the compressed output to a physical file on disk.
markup	A tool that uses a compression-based method for annotating a text file by inserting tags automatically (e.g. for code-switching, marking where the distinct languages occur in the text).
segment	A compression-based application used to perform word segmentation on a text file.
train	An application used to build the compression-based language models by "training" on text. These models are then used by the other applications in the toolkit. Two main types of models are possible: dynamic models, which are updated as the application proceeds (mainly used for adaptive encoding and decoding operations); and static models, which are not updated once they have been created.

The rest of this chapter is organized as follows. The next section discusses the three Welsh corpora used in the experimental evaluation which is discussed in Section 3. The experiments investigate the application of the TAWA toolkit to the following NLP tasks: compression; text classification; and sentence alignment for parallel corpora. The final section provides a discussion and future work.

2 EXPERIMENTAL CORPORA

In order to use TAWA most effectively, it is often helpful to use text corpora in order to train the language models. For this study which involves the Welsh language, this leaves a finite number of options as it is a less-resourced language. For this, two possible choices were the CEG corpus [3] as well as the more recent CorCenCC corpus [4], [5] both of which have been designed for multiple purposes as balanced corpora to provide samples of use of the language. A third corpus, the UAGT-PNAW corpus [6], in contrast is a parallel corpus containing both Welsh and English text. This corpus was added to the experimental evaluation in order to determine the effectiveness of the TAWA sentence alignment tool.

A further description of the three corpora is provided below.

2.1 CEG Corpus (1,046,551 words; 6,169,422 characters)

The 6.2 MB Cronfa Electroneg o Gymraeg (CEG) corpus is an inter-disciplinary collection of over 1 million Welsh words collected from 500 samples of text of 2000 words of diverse contemporary prose. The samples include materials from the fields of novels and short stories, religious writing, children's literature, public lectures, newspapers and magazines, memories, academic writing, general administration pieces, and various materials in the fields of education, science, business, leisure activities and more. This corpus was used to create and maintain CySill [3].

2.2 CorCenCC Corpus (13,443,584 words; 65,478,526 characters)

The 65.4 MB National Corpus of Contemporary Welsh (CorCenCC) is a recent inter-disciplinary and multi-institutional project that has created a large-scale, CC-BY-SA licensed corpus of contemporary Welsh. This corpus is constructed of examples of spoken, written and/or e-language examples from real life contexts. The first Welsh corpus of this type of construction, it offers an accurate view of the Welsh language across a diverse set of contexts of use, such as private chat messages, group socialising, business as well as other work situations, in education, in a range of media, and discourse in public spaces. It includes examples of news headlines, personal and professional emails and correspondence, academic writing, formal and informal speech, blog posts and text messaging. The language data was collated from a variety of different speakers of Welsh, representative of all regions of Wales, as well as all ages, genders and occupations, and with a variety of linguistic backgrounds, to accurately represent the range of text types and of Welsh speakers found in Wales today [4], [5].

2.3 UAGT-PNAW Corpus (510,813 aligned sentence pairs)

This corpus is a 100 MB bilingual Welsh–English corpus from the Proceedings of the National Assembly for Wales [6]. It provides a set of co-translated text for both Welsh and English arranged as aligned sentence pairs.

3 EXPERIMENTAL RESULTS

Various experiments were performed using the TAWA applications and the three corpora described above to determine the most effective way of modelling and processing Welsh text using compression-based language models. The first set of experiments initially investigated which models and training corpora were the most effective at compression. Once this was determined, this knowledge was applied to other applications in the subsequent sets of experiments.

3.1 Compressing Welsh Text Using Dynamic Models

The first set of experiments involved purely adaptive compression of Welsh text by selecting dynamic models with the **encode** tool in TAWA. For these models, the symbol probability estimations are essentially empty at the start of the text being compressed, and are subsequently updated as the compression proceeds. Often, dynamic models can outperform static models as PPM can quickly adapt to the specific text being compressed, whereas static models do not have the ability to change the probability estimations as the text is being processed sequentially.

These initial experiments are crucial to find out which models are the most effective for Welsh so that they can then be applied to produce the best results for the other applications. In the first set of experiments, the two text corpora, CEG and CorCenCC, were compressed directly for different order PPMD [1] models, from order 6 down to order 2. The results are shown in Figures 2 and 3. The graphs plot the compression ratio (bpc for bits per character) versus bytes input on a log scale and reveal how well the compression proceeds as the text files are being processed.

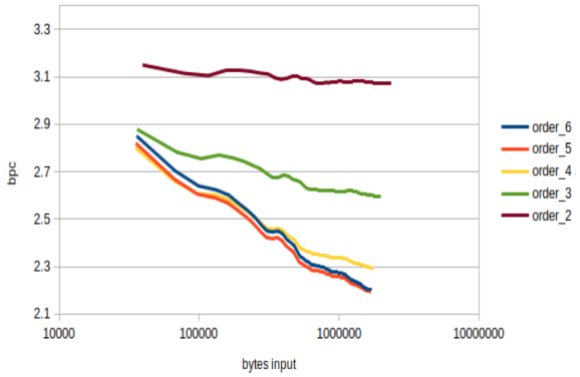

Figure 2: Encoding the CEG corpus dynamically.

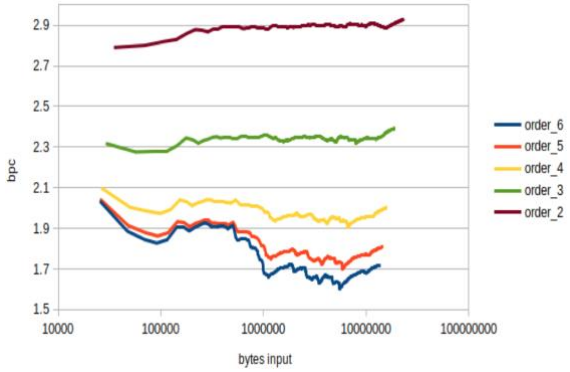

Figure 3: Encoding the CorCenCC corpus dynamically.

The results show that for the CEG corpus, compression generally improves consistently as the models are updated after more and more text is added. This is a standard result for natural language text and indicates that the language to the most part has consistent features i.e. the type of language does not vary significantly. There are significant improvements in compression from order 2 through to order 4 models. However, there is less improvement for orders 5 and 6 which perform at a similar level.

For the CorCenCC corpus, on the other hand, the picture is very different. There is consistent improvement in compression up until 0.10 Mbytes input, but then the compression noticeably decreases. The compression ratio does not drop again until about 0.70 Mbytes. Further notable rises in compression ratio occur at 1.0 and 7.5 Mbytes. This indicates that the corpus consists of various text genres from different domains that diverge substantially from each other throughout. As a consequence, although this corpus is much larger than the CEG corpus (being nearly 10.613 times bigger in size), using this corpus to train language models for statistical NLP may be less effective for compressing different testing text genres due to its more heterogeneous nature. Instead, improved performance in various NLP tasks may be better served by extracting the distinctive parts of the corpus and using them to build separate models.

3.2 Compressing Welsh Text Using Static Models

The second set of experiments explored the effectiveness of using a static model built from one corpus to potentially improve the compression of another corpus. In this case, a static model is primed from one corpus using TAWA's **train** tool, and this is then used to compress the other corpus using the **encode** tool. This was again done for orders 6 down to 2 using PPMD models. The results of the experiments are shown in Figures 4 and 5.

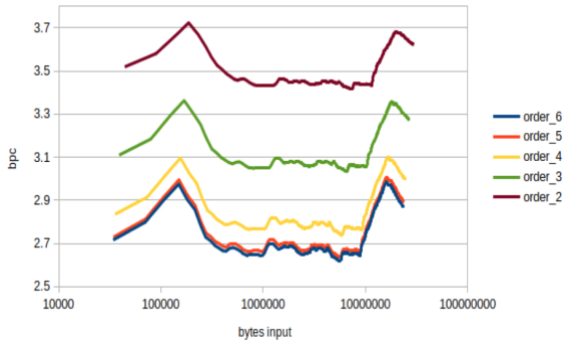

Figure 4: Encoding CorCenCC statically using CEG for training.

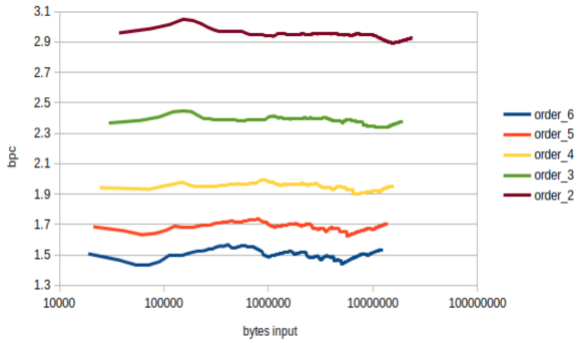

Figure 2: Encoding CEG statically using CorCenCC for training.

The results are very different to the results in Figures 2 and 3. At first glance, they seem to show that there is no reason to use a training model as the compression algorithm performs worse statically. As stated above, it is known that the dynamic model can outperform the static model given sufficient training data as the model is adaptive and updates itself as the text is being processed.

In this case, using CEG as the training corpus is unwise when encoding CorCenCC (see Figure 4) as the particular static model is not a good predictor of the text being predicted at two main points, the sharp peaks that occur at roughly 200,000 and 20,000,000 bytes. This is most likely down to changes in the text due to different genres or styles occurring at those points.

The advantage of using a pre-primed static model is that the compression can be effective immediately from the start, but only if the training text is representative of the text being compressed. For example, the compression ratio starts at 1.5 bpc for the order 6 model when encoding the CEG corpus (see Figure 5) rather than at 2.8 bpc in Figure 2. In contrast, the compression ratio starts at 2.7 bpc in Figure 4 which is worse than the 2.0 bpc in Figure 3. This indicates that the CEG text is not a good predictor of CorCenCC.

An important conclusion from Figures 2 to 5, however, is that for both dynamic and static encoding, order 6 models yield the best results for Welsh.

3.3 Text Classification of Welsh Text

Compression-based text classification has been shown to be competitive in many different domains [1]. The purpose of this preliminary experiment is to see how well it performs at classifying Welsh text. The documents in the CEG corpus, in particular, have been manually classified by text genre so these can be used as the 'ground truth' for this experiment. Specifically, the documents have been classified into two main categories – 'Fiction' and 'Non-Fiction', as well into 21 sub-categories such as 'Press – Scientific', 'Academic', 'Novels' and 'Short Stories'.

For this preliminary experiment, we decided not to use the sub-categories as many of these do not have sufficient training documents for training the PPM models. We used 5-fold cross-validation for a robust evaluation across the whole dataset. Test files were classified in each fold using static PPMD models built using training data from the rest of the dataset. The results are shown below using different order PPMD models (from order 2 to order 6). *TP*, *FP*, *FN* and *TN* are the number of true positives, false positives, false negatives and true negatives respectively when considering the task as a binary classification problem, with 'Fiction' being the positive class and 'Non-Fiction' being the negative class. Accuracy is calculated as (*TP* + *TN*) / (*TP* + *TN* + *FP* + *FN*), recall as *TP* / (*TP* + *FN*), precision as *TP* / (*TP* + *FP*) and F1 Score as (2 * *TP*) / (2 * *TP* + *FP* + *FN*).

Table 2: Results of preliminary experiment

Order	TP	FP	FN	TN	Accuracy	Recall	Precision	F1 Score
2	89	43	15	350	0.883	0.856	0.674	0.754
3	89	32	15	361	0.905	0.856	0.736	0.791
4	75	13	29	380	0.915	0.721	0.852	0.781
5	58	6	46	387	0.895	0.558	0.906	0.690
6	40	4	64	389	0.863	0.385	0.909	0.541

The results show that the compression-based classifier performs with a high degree of accuracy (0.915), especially when order 4 models are used. Recall is highest with order 2 and 3 models (0.856), Precision is highest with order 6 models (0.909) and F1 Score is highest with order 3 models (0.791). The results indicate that the compression-based method is very effective at distinguishing between fictional and non-fictional Welsh text.

Other experimental results in different languages and domains (such as genre and gender) show that compression-based models are competitive with or outperform state-of-the-art machine learning methods. A full experimental comparison with these methods has been left for future work.

3.4 Checking Sentence Alignment in Parallel Welsh-English Text Using Compression Codelength

This section reports on some experiments at checking the sentence alignment for the UAGT-PNAW corpus. This is done using compression codelength to measure the information content in each text character string, estimated using the **codelength** tool in the TAWA toolkit. The insight behind this approach hinges on the premise that the compression of co-translated text (i.e. sentences in this case) should have similar code lengths [1] since the information contained in the co-translations will be similar. Since compression can be used to measure the information content, we can simply look at the ratio of the compression code lengths of the co-translated text pair to determine whether the text is aligned. That is, if you have a sentence text string in one language, and its translation in another language, then the ratio R of the compression code lengths of the text string pair should be close to 1.0, where $R = \max(C_W/C_E\,, C_E/C_W)$ and C_W is the compression codelength for the Welsh sentence and C_E is the compression codelength for the English sentence.

Results of preliminary experiments for the UAGT-PNAW corpus are shown in Figures 6 to 9. Note that compression codelength in this case is calculated using pre-primed static models trained on text from the respective languages because the sentence texts are too short for dynamic PPM models to produce distinguishable results. The CEG corpus is used to train the Welsh model for compressing the Welsh sentences, and the LOB English corpus is used to train the English model for compressing the English sentences.

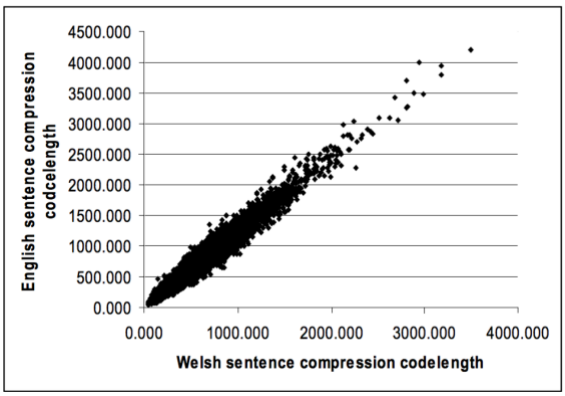

Figure 6: Compression codelengths for the first 10,000 sentences in the UAGT-PNAW parallel corpus.

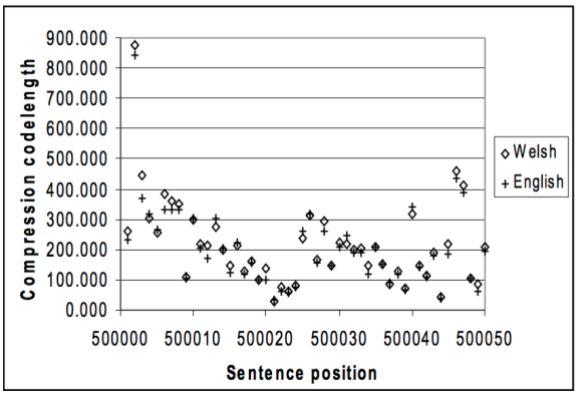

Figure 7: Compression codelengths for sentences in the UAGT-PNAW parallel corpus.

Figure 6 graphs compression codelength ratios versus sentence position for the corpus. The actual codelengths for both the Welsh and English sentences are plotted for sentence positions 500,000 through 500,050 in Figure 7. The figures clearly show that the compression codelengths for most sentences correlate extremely well being very similar in value.

Figure 8 plots the compression codelengths ratios *R* for the same sentence positions 500,000 through 500,050 for the corpus. As stated, these ratios should be close to 1.0. This is the case until sentence position 501,719 when the ratio suddenly jumps to over 10.0. After this, all the following sentences are clearly mis-aligned. Once this mistake is corrected, the following sentences then become correctly aligned except for the obvious further possible error at around sentence position 503,000.

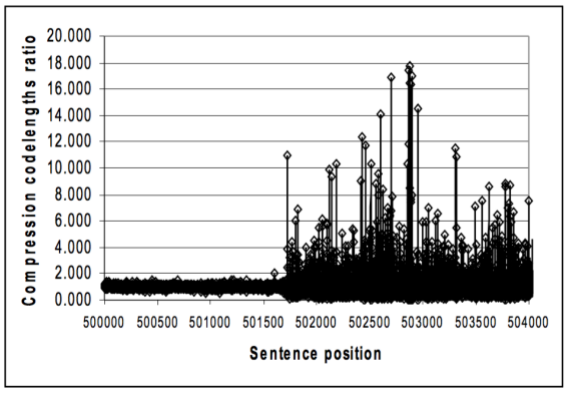

Figure 8: Compression codelengths ratios for sentences in the UAGT-PNAW parallel corpus.

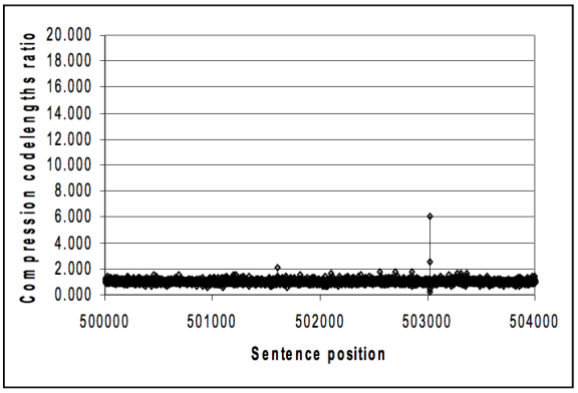

Figure 9: After correcting error at sentence position 501719

DISCUSSION AND FUTUE WORK

This chapter has investigated the application of the compression-based TAWA toolkit to various problems in Welsh NLP. Three Welsh corpora were used in the experimental evaluation: the CEG corpus [3]; the CorCenCC corpus [4], [5] which has recently become available; and the UAGT-PNAW Welsh-English bilingual corpus [6]. The results show th at the tools in the toolkit can be applied to produce effective performance for three tasks: compression; text categorization; and sentence alignment for parallel texts. This work serves as an important foundation on which to formally apply the compression-based NLP methods for the Welsh language.

Many more applications are possible and will be investigated in future work. This will include investigating the automatic detection of code-switching using TAWA's **markup** tool as the CEG corpus has examples of Welsh toEnglish code switching as well as Welsh to French. Additionally, it would be interesting to update CySill using the CorCenCC corpus as it provides a far larger data-set to train on.

The importance of considering the linguistic diversity of the Welsh language cannot be underestimated. Using CEG as a training model for encoding the CorCenCC served as a good lesson in the quality of the results versus the size and diversity of the corpus. However, the results using CorCenCC as a training model for encoding the CEG corpus were impressive.

REFERENCES

[1] William J. Teahan. 2018. A Compression-Based Toolkit for Modelling and Processing Natural Language Text. Information, 9, 12 (November 2018). http://doi:10.3390/infoxx010001.

[2] John G. Cleary and Ian H.Witten. 1984. Data compression using adaptive coding and partial string matching. IEEE Transactions on Communications, 32, 4 (April 1984), 396-402.

[3] Nick C. Ellis, C. O'Dochartaigh, William Hicks, M. Morgan and N. Laporte. 2001. Cronfa Electroneg o Gymraeg (CEG): A 1 million-word lexical database and frequency count for Welsh. [On-line]

[4] Dawn Knight, Steve Morris, Tess Fitzpatrick, Paul Rayson, Irena Spasić, Enlli-Mon Thomas, Alex Lovell, Jonathan Morris, Jeremy Evas, Mark Stonelake, Laura Arman, Josh Davies, Ignatius Ezeani, Steve Neale, Jennifer Needs, Scott Piao, Mair Rees, Gareth Watkins, Lowri Williams, Vignesh Muralidaran, Bethan Tovey-Walsh, Laurence Anthony, Thomas M. Cobb, Margaret Deuchar, Kevin Donnelly, Michael McCarthy and Kevin Scannell. 2020. CorCenCC: Corpws Cenedlaethol Cymraeg Cyfoes – the National Corpus of Contemporary Welsh Dataset. http://doi.org/10.17035/d.2020.0119878310.

[5] Dawn Knight, Steve Morris, Tess Fitzpatrick, Paul Rayson, Irena Spasić and Enlli-Mon Thomas. 2020. The National Corpus of Contemporary Welsh: Project Report | Y Corpws Cenedlaethol Cymraeg Cyfoes: Adroddiad y Prosiect. Cardiff University, Wales. arXiv:2010.05542.

[6] Dafydd Jones and Andreas Eisele. 2006. Phrase-based statistical machine translation between English and Welsh. In Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages), LREC-2006. LREC, Genoa, Italy, 75-78.

Bilingual Welsh and English Text-to-Speech

DEWI BRYN JONES

Language Technologies Unit, Bangor University, Wales

DR SARAH COOPER

School of Arts, Culture and Language, Bangor University, Wales

This chapter describes work on the development of a bilingual Welsh-English text-to-speech system by combining existing Welsh and English pronunciation dictionaries and by improving the quality of a small bilingual speech corpus. The new bilingual pronunciation dictionary was expanded with new words with the help of new grapheme-to-phoneme models which were trained and evaluated for their accuracy. The voice was built with the unit selection module of MaryTTS which, according to preliminary listening alone, produces intelligible Welsh and English language speech. However, some limitations are highlighted along with a discussion on further work.

1 INTRODUCTION

Text-to-speech systems enable computers to produce audio versions of text in a specific language within a wide range of applications such as web page readers, speech-improved user interfaces, digital personal assistants and automated publishing systems [1], [2].

For some years, a variety of systems have been producing audio from Welsh texts. Welsh text-to-speech was created on a biphone basis with the Festival speech synthesis system [2] in 2006 as a result of collaboration between Bangor University and Dublin universities during an EU Interreg project entitled WISPR (Welsh and Irish Speech Processing Resources). WISPR also developed a number of speech resources for the further research purposes of Welsh and Irish language speech technology [4]. In 2009 a commercial Welsh text-to-speech product was published by the Ivona company that had a much more natural synthesis quality. The provision is now available within Amazon Polly's speech service. Later in 2016 an open source voice that also sounded more natural was developed with the MaryTTS system [5] for the use of a Welsh personal digital assistant - Macsen [6] and a speech banking programme called Lleisiwr.[1]

However, Welsh language text-to-speech users live and communicate in bilingual environments that also need to produce audio from English texts. Using two separate single-language text-to-speech systems and switching between them is neither practical nor adequate for pronunciation of borrowed words, some proper nouns or of instances of code switching. As a result, a single text-to-speech system, capable of producing naturally-sounding voice audio for texts of both languages, is highly desirable. Purchase and maintenance of a single text-to-speech system, rather than two, can benefit organisations requiring public address systems and users of the Lleisiwr programme who benefit from the convenience of a more effective banking and training process.

[1] https://lleisiwr.cymru

2 RESOURCE ANALYSIS

The speech synthesis solution method influences what resources are required. It was decided to create an initial bilingual text-to-speech synthesis with the open resources from the work already done with Festival and MaryTTS. The unit selection module of MaryTTS, which requires phonetic analysis of language, namely a pronunciation dictionary, and high-quality recordings of an actor reading a specially designed recording script, was used.

2.1 Pronunciation Dictionaries

As we wanted a bilingual text-to-speech system we needed to create a bilingual pronunciation dictionary containing large numbers of words from both languages described phonetically with a series of phonemes. This required one set of phonemes to represent each possible sound from both languages.

It was decided to construct a bilingual dictionary by combining the words and phonemes from open dictionaries already available in English and Welsh on the basis of emulating methods of forming a superset of phonemes used by others [7]. The Festival Welsh voice pronunciation dictionary was chosen along with the American English CMU Pronouncing Dictionary.[2]

2.1.1 Festival Welsh Voice Pronunciation Dictionary

The Festival Welsh voice pronunciation dictionary includes phonetic transcripts for almost 39000 words with a phoneme set consisting of 14 short and long vowels, 35 consonants and 14 diphthongs.

A sample transcript can be found in table 1. Phonemes are represented as ASCII characters according to the Machine-Readable Phonetic Alphabet for Welsh (MRPAW) as designed in [8]. Syllables are enclosed in brackets. An emphasis is indicated with the number 1 whilst 0 indicates that there is no emphasis.

The dictionary has grown in size very slowly over a number of years with words added and transcribed manually. Analysis of the dictionary by comparison with the Welsh and English Hunspell[3] spell checker glossaries revealed that it contained around 26000 words in Welsh, 12000 words in English, 2600 in both languages whilst approximately 3500 were not recognised as Welsh or English words. The list of Welsh words includes lemmas, mutations and inflexions. Not all inflexions of lemma are present. The dictionary contains an additional field for word tagging with one of 61 possible meta-data elements, as part of a phrase. But the additional field does not record the language of the word. The 'foreign' tag identifies borrowed words from English or spelling according to Welsh orthography of an original English word e.g. 'byjamas', 'telefision'. Otherwise, English words use the same phrase parts as Welsh words.

The existence and number of English words suggests that there was an effort to enable Welsh Festival voices to pronounce at least a limited set of words in English from the Welsh language set of phonemes – although one additional phoneme has been included for noting a vocal post-alveolar fricative (zh /ʒ/) in words such as 'explosion', 'measure' and 'television'.

[2]The CMU Pronouncing Dictionary, available from http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[3]Hunspell Cymraeg, available from http://techiaith.cymru/data/geiriadurol/hunspell-cymraeg/

We do not know the basis for adding and phonetically transcribing these English words. The number is relatively small compared to the needs of a full pronunciation dictionary. We cannot expect this to be adequate for pronouncing a wide range of English texts within bilingual text-to-speech voices.

2.1.2 CMU Pronunciation Dictionary

The CMU pronunciation dictionary (CMUDict) is an open source pronunciation dictionary created by the speech research group at Carnegie Mellon University. It includes phonetic transcripts of North American pronunciations of English words. A sample transcript can be found in table 1.

Phonemes are represented as ASCII characters using ARPABET codes.[4] An additional field, which includes a single number only, is used to indicate an alternative pronunciation for a word. CMUDict does not note any information regarding syllables. Emphasis is noted by attaching 1 (main emphasis) or 2 (second emphasis) to the relevant phoneme.

In the latest version - 0.7b (published 2014) - CMUDict contains approximately 138,000 records.

Table 1 – Comparison of examples of Welsh and English pronunciation dictionaries

Word	Pronunciation Dictionary	Phonetic Description
Text-to-speech	Festival (Welsh)	adj (((t e s) 0) ((t y n) 0) ((ii l) 0) ((e v) 0) ((e r) 1) ((i dh) 0))
Text-to-Speech	CMUDict (English)	T EH1 K S T . T UW1 . S P IY1 CH .

CMUDict has been used as a training corpus for the construction of grapheme-to-phoneme (G2P) models that can produce phonetic descriptions for new words that are not in the dictionary.

2.2 WISPR Corpus Speech Data

The WISPR project [4] produced high quality studio speech recordings intended as a selected speech synthesis training tool for Welsh language units, despite the project focus being on realising biphone speech synthesis. It is now available from the language technologies portal.[5] This work is the first major attempt to use it for text-to-speech training.

There is little documentation or metadata in the corpus. The corpus has recordings of 265 sentences, some of them being very long, from the Welsh Bible, as well as 351 recordings of sentences from an undergraduate student's dissertation. Some sentences are wholly or partly in English. In total, there are 3 hours and 30 minutes of recordings, with 20 minutes of recordings of full English sentences. The voice talent is male and has a North Walian accent.

3 REALIZING BILINGUAL TEXT-TO-SPEECH

3.1 Creating a Phonemes Superset

All phonemes from both dictionaries were translated into their standard equivalent phonemes from the IPA.[6]

[4]ARPABET - https://en.wikipedia.org/wiki/ARPABET

[5]WISPR Corpus - http://techiaith.cymru/data/corpora/wispr/

[6]The International Phonetic Association - https://www.internationalphoneticassociation.org/

As a result, we realised that MRPAW and ARPABET consonant phonemes corresponded very well, but there were more significant differences between the phonemes of vowels and diphthongs. As MRPAW contains a greater variety of phonemes we were able to connect CMUDict unique phonemes to phonemes judged similar enough in MRPAW rather than forming a simple superset. MRPAW was inadequate to identify just one unique CMUDict phoneme, namely the rhotic vowel (/ɜ˞/) as heard in the words 'hurt', 'mirror' and 'water'.

MRPAW was originally designed to transcribe the Welsh language with a South Walian accent. As a result, it consisted of three labialized consonants (/lw/, /nw/ and /rw/). These were judged to be negligible and irrelevant to other accents and for Welsh-only text-to-speech in general. Other studies in Welsh language phonology ([9]; [10]; [11]) also do not include labialized consonants.

See the appendix for each final set of phonemes and their connection to Welsh Festival and CMUDict phonemes.

3.2 Creating a Balanced Bilingual Pronunciation Dictionary

Both dictionaries are very different in size, with CMUDict containing nearly 100k words more than the Festival Welsh dictionary. In order to avoid a trend towards one language within any bilingual text-to-speech system which would cause words to be pronounced incorrectly, a balance of the same number of words from both languages was sought. This was achieved by increasing the number of Welsh words and reducing the number of English words to a collection of most common words.

Unfortunately, English word frequency lists within huge contemporary corpora are neither readily available, nor freely available with a permitted licence to create a list of more common words, although lists are available to purchase with unsuitable redistribution restrictions [12]. We produced an English word frequency list ourselves with the open and accessible resources of the NLTK library [13] and the Brown corpus [14]. We also used a list of 500 written English words that are used most often in Welsh [15]. As a result, the balanced bilingual pronunciation dictionary contains up to 20000 English words and up to 30000 Welsh words.

A solution was needed to balance word syllable information. The Festival Welsh pronunciation dictionary denotes syllables within brackets, whilst CMUDict contains no information at all. We found and used the results of work [16] on using sonority sequencing methods to produce syllables for our selection of words from a CMUDict dictionary.

Sample records can be found below in figure 1. The format provides four columns that are easy to parse on a computer basis:

1. The word or the graphemes
2. Metadata, enclosed by brackets, for specifying language, part of speech or source
3. Phonemes in the form of ASCII characters (MRPAW column in the appendix table)
4. Phonemes in the form of characters from the IPA alphabet, enclosed by forward slash

Phonemes are separated by a gap, syllables by a hyphen, whilst an apostrophe indicates the position of emphasis.

```
syml 's @ - m y l /'sə.mɨl/
symlaf 's @ m - l a v /'səm.lav/
symleiddio s @ m - 'l ei dh - j o /səm.'ləɪð.jɔ/
symlrwydd 's @ m l - r uy dh /'səml.rʊɨð/
symol 's @ - m o l /'sə.mɔl/
symposiwm s @ m - 'p oo - s iu m /səm.'pɔː.sɪʊm/
symptomau s @ m p - 't o - m ay /səmp.'tɔ.maɨ/

symphonies (en,cmu) s 'ɪ m - f @ - n ii z /s'ɪm.fə.niːz/
symphony (en,cmu) s 'ɪ m - f @ - n ii /s'ɪm.fə.niː/
symposium (en,cmu) s ɪ m - p 'ou - z ii - @ m /sɪm.p'əʊ.ziː.əm/
symptom (en,cmu) s 'ɪ m p - t @ m /s'ɪmp.təm/
symptomatic (en,cmu) s 'ɪ m p - t @ - m 'a - t ɪ k /s'ɪmp.tə.m'a.tɪk/
symptoms (en,cmu) s 'ɪ m p - t @ m z /s'ɪmp.təmz/
synagogue (en,cmu) s 'ɪ - n @ - g 'oo g /s'ɪ.nə.g'ɔːg/
```

Figure1 - Examples of records from the bilingual pronunciation dictionary

3.3 G2P Model Training

Welsh words needed to be added to the pronunciation dictionary so a variety of grapheme-to-phoneme (G2P) models were trained to avoid transcribing each one phonetically by hand. It was decided in this work to experiment with different collections of training data and not to evaluate different methods of modelling G2P. Only Phonetisaurus [17], which is the de-facto open source library, was used.

The results of our experiments are shown in table 2 below. A 'k-fold cross validation' testing method was used to split the small amount of data into the best test and training sets to accurately measure the model's ability to accurately transcribe new words phonetically. Accuracy was measured with the word error rate metric (WER) which is the number of incorrect phonetic transcriptions divided by the total number of words in the test set.

We see from these results that specific language G2P models transcribe phonology better than the bilingual models. As a result, the Festival Welsh model was used to facilitate the addition of Welsh words to the pronunciation dictionary and the CMUDict model (entire dictionary) for new English words.

Table 2 – Results of G2P models

Training Set	Size	WER
Festival Welsh	28764	25.76%
CMUDict (most frequently used words set)	20622	35.34%
Festival Welsh + CMUDict (most frequently used words set)	49386	34.30%
CMUDict (entire dictionary)	119305	32.65%

3.4 WISPR Speech Data Corpus Corrections

Our analysis of the WISPR speech corpus identified some problems and errors which may undermine its usefulness for training a text-to-speech system.

Spelling mistakes were corrected as well as removing excessive punctuation. A considerable number of sentences in the undergraduate dissertation included numbers, such as a year, in the form of digits. These were all converted into word form as pronounced by the voice talent in the recordings.

Finally, the Welsh and English vocabulary of all the sentences was collected from the corpus in order to add them, with the aid of the best G2P models, to the bilingual pronunciation dictionary.

3.5 Training MaryTTS

MaryTTS [5] is a popular text-to-speech synthesis library among academic researchers and open source and commercial software developers alike. As we have already used MaryTTS to inform the development of Welsh only text-to-speech, resources and components were available to use again for the development of the first bilingual Welsh-English synthesis.

The XML file (`allophones.cy.xml`) which was already available in Welsh in MaryTTS' provision was expanded to particularize phonemes within the new bilingual pronunciation dictionary. Further information was added regarding the audio features of all the phonemes such as loudness, type of consonant, location of expression and vocalization. See example pieces in the appendix to this chapter and the entire file at GitHub.[7]

With the new bilingual pronunciation dictionary and data from the WISPR corpus having been improved, MaryTTS could be trained to create the bilingual Welsh-English text-to-speech.

4 VOICE EVALUATION

In this work, it was only possible to carry out preliminary listening of the new bilingual Welsh-English voice. But it was noted that:

- the Welsh and English language speech produced by MaryTTS was intelligible.
- it was possible to produce intelligible speech from the long and challenging sentences of Welsh and English news articles and literature.
- rarely were there missing phonemes or words in the speech.
- pronunciation of examples of language mixing within text, such as proper names and code switching, were correct for the most part.
- translanguage homographs, such as 'wall' and 'plant' were often pronounced in the wrong language.

It is possible to experiment with the voice on the text-to-speech page of the language technologies portal website.[8]

5 CONCLUSION

This chapter has reported the way in which bilingual Welsh-English text-to-speech was created through improvements to existing open source text-to-speech resources and the MaryTTS library.

The new bilingual voice was, following a preliminary listening exercise, sufficiently intelligible and beneficial. It has therefore already been adopted within the products and services of the Language Technologies Unit such as Lleisiwr and Macsen. However, a detailed evaluation with the aid of human listeners is required in order to identify a Mean Opinion Score (MOS) and to compare it with monolingual voices.

[7] https://github.com/techiaith/marytts/blob/marytts-lang-cy/marytts-languages/marytts-lang-cy/lib/modules/cy/lexicon/allophones.cy.xml

[8] Demo of the bilingual voice on the Language Technologies Portal website - http://techiaith.cymru/lleferydd/testun-i-lleferydd/

A new bilingual pronunciation dictionary was developed as well as the best possible G2P models in order to facilitate the addition of new words. A simple analysis found that around 2600 translanguage homographs existed between the two languages and that the new bilingual voice often pronounced them incorrectly. Further work is needed on confirming the language of words within texts.

WISPR speech corpus quality was improved and made more useful to everyone.

The intention is to create more bilingual Welsh-English text-to-speech voices with larger collections of recordings from 4 new voice talents (2 males with north Walian and south Walian accents, 2 females with north Walian and south Walian accents) with the aim of increasing the range of voices and improving the natural quality of text-to-speech voices. The recording scripts have been designed for them with the help of this work's new bilingual pronunciation dictionary.

References

[1] Daniel Jurafsky and James H. Martin. 2014. Speech and language processing (2nd. ed., Pearson new internat. ed.). Pearson Education International. Harlow.

[2] Dewi Bryn Jones, Delyth Prys, Myfyr Prys, and Gruffudd Prys. 2019. Llawlyfr technolegau iaith. Coleg Cymraeg Cenedlaethol. Bangor, Wales.

[3] Alan W. Black, and Kevin A. Lenzo. 2000. Building voices in the Festival speech synthesis system. Retrieved September 5, 2020 from http://www.festvox.org/festvox-1.1/

[4] Briony Williams, Rhys J.Jones, and Ivan Uemlianin. 2006. Tools and resources for speech synthesis arising from a Welsh TTS project. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006). European Language Resources Association, Genoa, Italy, 2574-2577.

[5] Mark Schröder and Jürgen Trouvain. 2003. The German text-to-speech synthesis system MARY: A tool for research, development and teaching. In International Journal of Speech Technology 6, 4 (October 2003), 365-377. https://doi.org/10.1023/A:1025708916924

[6] Dewi Bryn Jones. 2020. Macsen: A Voice Assistant for Speakers of a Lesser Resourced Language. In Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL). European Language Resources Association, Marseille, France, 194-201.

[7] Slavcho Chungurski, Ivan Kraljevski, Igor Stojanovic and Blerta Prevalla. 2010. Speech Synthesis of Dissimilar Languages Using their Phonetic Superset. In Proceedings of 8th Conference, Digital Speech and Image Processing. Iriski Venac, Serbia, A 4.8.

[8] Briony Williams. 1994. Welsh letter-to-sound rules: Rewrite rules and two-level rules compared. Computer Speech and Language, vol. 8 (July 1994), 261-277.

[9] Robert Mayr and Hannah Davies. 2011. A cross-dialectal acoustic study of the monophthongs and diphthongs of Welsh. Journal of the International Phonetic Association (April 2011), 41 1 - 25.

[10] Glyn E. Jones. 1984. The distinctive vowels and consonants of Welsh. In Welsh phonology: Selected readings. University of Wales Press, Cardiff, 40-64.

[11] Sarah Cooper and Laura Arman (editors). 2020. Cyflwyniad i ieithyddiaeth, Coleg Cymraeg Cenedlaethol. Y Coleg Cymraeg Cenedlaethol, Caerfyrddin. ePDF 978-1-911528-19-7

[12] Mark Davies. 2011. Most frequent 100,000 word forms in English (based on data from the COCA corpus). Retrieved September 5, 2020 from https://www.wordfrequency.info/.

[13] Steven Bird, Edward Loper and Ewan Klein. 2009. Natural Language Processing with Python. O'Reilly Media Inc.

[14] Winthrop N. Francis and Henry Kučera. (1964, 1971, 1979) A Standard Corpus of Present-Day Edited American English, for use with Digital Computers (Brown). Brown University. Providence, Rhode Island.

[15] Delyth Prys and Dewi Bryn Jones. 2019. Most common wordforms in Welsh and the most common English words used in Welsh. Retrieved September 5, 2020 from http://techiaith.cymru/data/lexicons/wordlists-of-the-most-common-wordforms-in-welsh-and-the-most-common-english-words-used-in-welsh/?lang=en

[16] Susan Bartlett, Grzegorz Kondrak and Colin Cherry. 2009. On the Syllabification of Phonemes. In Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings. Association for Computational Linguistics, Boulder, Colorado, USA, 308-316.

[17] Josef Novak, Nobuaki Minematsu and Keikichi Hirose. 2015. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework. Natural Language Engineering 22, 6 (November 2016), 907-938. DOI: 10.1017/S1351324915000315.

APPENDIX

Welsh-English Bilingual Pronunciation Phonemes Superset

Table 1: Llafariaid / Vowels

MRPAW	IPA	Description	Example	CMU
a	a	Llafariad isel agored blaen Open front unrounded vowel		AE /æ/ (at)
aa	aː	Llafariad isel agored blaen hir Open front unrounded vowel long	sâl, tad	
e	ɛ	Llafariad canolog blaen Open-mid front unrounded vowel		EH
ee	eː	Llafariad canolog blaen hir Open-mid front unrounded vowel long	mêl	
i	ɪ	Llafariad uchel caeedig blaen Near-close near-front unrounded vowel		IH
ii	iː	Llafariad uchel caeedig blaen hir Near-close near-front unrounded vowel long	ci	IY /i/ (eat)
o	ɔ	Llafariad canolog cefn Open-mid back rounded vowel		AA /ä/ (odd)
oo	ɔː	Llafariad canolog cefn hir Open-mid back rounded vowel long	tôn	AO /ɔ/ (ought)
u	ʊ	Llafariad uchel caeedig cefn Close back rounded vowel	cwm	UH
uu	uː	Llafariad uchel caeedig cefn hir Close back rounded vowel long	cŵn, stŵr	UW /u/ (two)
@	ə	Llafariad canolog canol (shwa) Mid central vowel (schwa)		AH /ɐ/ (hut)
@@	ɜː	Llafariad canolog blaen hir Open-mid central unrounded vowel long	word	
yy	ɨː	Llafariad uchel caeedig canol hir Near-close back rounded vowel long	sych, tŷ	
@r	ər	Llafariaid rhotig (en_US) Rhotic vowel (en_US)	mirror	ER /ɝ/ (hurt)

Table 2: Cytseiniau / *Consonants*

MRPAW	IPA	Description	Example	CMU
b	b	Ffrwydrolyn dwywefusol lleisiol Voiced plosive bilabial		B
k	k	Ffrwydrolyn felar di-lais Voiceless velar plosive		K
x	χ	Ffrithiolyn wfwlar di-lais Voiceless uvular fricative		
d	d	Ffrwydrolyn gorfannol lleisiol Voiced alveolar plosive		D
dh	ð	Ffrithiolyn deintiol lleisiol Voiced dental fricative		DH
v	v	Ffirthiolyn gwefus-ddeintiol lleisiol Voiced labiodental fricative		V
f	f	Ffrithiolyn gwefus-ddeintiol ddi-lais Voiceless labiodental fricative		F
g	g	Ffrwydrolyn felar lleisiol Voiced velar plosive		G
hh	h	Ffrithiolyn glottal di-lais Voiceless glottal fricative		HH
jh	dʒ	Affrithiolyn ôl-orfannol lleisiol Voiced postalveolar affricate	Jones, John	JH
l	l	Dynesolyn ochrol gorfannol lleisiol Voiced alveolar lateral approximant		L
lh	ɬ	Ffrithiolyn ochrol gorfannol di-lais Voiceless alveolar lateral fricative		
m	m	Trwynolyn dwywefusol lleisiol Voiced bilabial nasal		M
mh	m̥	Trwynolyn dwywefusol di-lais Voiceless bilabial nasal		
n	n	Trwynolyn gorfannol lleisiol Voiced alveolar nasal		N
nh	n̥	Trwynolyn gorfannol di-lais Voiceless alveolar nasal		
ng	ŋ	Trwynolyn felar lleisiol Voiced velar nasal		NG
ngh	ŋ̊	Trwynolyn felar di-lais Voiceless velar nasal		
p	p	Ffrwydrolyn dwywefusol di-lais Voiceless plosive bilabial		P
r	r	Tril gorfannol lleisiol Voiced alveolar trill		R /ɹ/ (read)
rh	r̥	Tril gorfannol di-lais Voiceless alveolar trill		
s	s	Ffrithiolyn gorfannol di-lais Voiceless alveolar fricative		S
sh	ʃ	Ffrithiolyn ôl-orfannol di-lais Voiceless postalveolar fricative		SH
t	t	Ffrwydrolyn gorfannol di-lais Voiceless alveolar plosive		T

MRPAW	IPA	Description	Example	CMU
th	θ	Ffrithiolyn deintiol di-lais Voiceless dental fricative		TH
ch	tʃ	Affrithiolyn ôl-orfannol di-lais Voiceless postalveolar affricate	cwtsh	CH
z	z	Ffrithiolyn gorfannol lleisol Voiced alveolar sibilant	crazy	Z
zh	ʒ	Ffrithiolyn ôl-orfannol lleisiol Voiced palato-alveolar fricative	explosion	ZH
j	j	Dynesolyn taflodol lleisiol Voiced palatal approximant	wincio	Y
w	w	Dynesolyn felar lleisiol Voiced labial-velar approximant	swigen	W

Table 3: Deuseiniaid / *Diphthongs*

MRPAW	IPA	Description	Example	CMU
ai	aɪ	Deusein cau blaen Front closing diphthong	llais, tai	AY
ei	əɪ	Deusein cau blaen Front closing diphthong	tei, neidio	EY /eɪ/ (ate)
eu	ɛʊ	Deusein cau blaen Front closing diphthong	mewn, llew	
oi	ɔɪ	Deusein cau blaen Front closing diphthong	cnoi	OY
ay	aɨ	Deusain cau canolog Central closing diphthong	haul, cau	
aay	ɑɨ	Deusain cau canolog Central closing diphthong	cae	
ey	əɨ	Deusain cau canolog Central closing diphthong	creu, neu	
oy	ɔɨ	Deusain cau canolog Central closing diphthong	troed, coed	
uy	ʊɨ	Deusain cau canolog Central closing diphthong	dwyn, mwy	
iu	ɪʊ	Deusain cau cefn Back closing diphthong	briw	
ou	əʊ	Deusain cau cefn Back closing diphthong	trowsus	OW /oʊ/ (oat)
yu	ɨʊ	Deusain cau cefn Back closing diphthong	clyw Duw	
au	aʊ	Deusain cau cefn Back closing diphthong	pawb, llaw	AW

Sample pieces from MaryTTS' allophones.cy.xml file

```
<allophones name="bangor-mrpaw"
            xml:lang="cy"
        features="vheight vfront vrnd ctype cplace cvox">

    <silence ph="_"/>

        <!-- ############################ -->
        <!-- Monophthongs / Unseiniaid   -->
        <!-- ############################ -->

    <!-- open front unrounded vowel /a/ -->
    <vowel ph="a" vc="+" vlng="0" vheight="3" vfront="1" vrnd="-"/>

    <!-- open front unrounded vowel (long) /aː/ 'sâl, tad' -->
    <vowel ph="aa" vc="+" vlng="l" vheight="3" vfront="1" vrnd="-"/>

    <!-- open mid-front unrounded vowel /ɛ/ -->
    <vowel ph="e" vc="+" vlng="0" vheight="2" vfront="1" vrnd="-"/>


        <!-- ############################-->
        <!-- Diphthongs / Deuseiniaid    -->
        <!-- ############################-->

        <!-- Front closing -->

        <!-- /aɪ/  'llais, tai'  -->
        <vowel ph="ai" vc="+" vlng="d" vheight="1" vfront="1" vrnd="-"/>

        <!-- /əɪ/ 'tei, neidio' -->
        <vowel ph="ei" vc="+" vlng="d" vheight="1" vfront="1" vrnd="-"/>


        <!-- ######################################## -->
        <!-- Consonants     /     Cytseiniaid       -->
        <!-- ######################################## -->
        <!--

    <!-- b /b/ voiced stop/plosive bilabial -->
        <consonant ph="b" vc="-" ctype="s" cplace="l" cvox="+"/>

        <!-- k /k/ voiceless velar stop/plosive -->
        <consonant ph="k" vc="-" ctype="s" cplace="v" cvox="-"/>

    <!-- x /χ/ voiceless uvular fricative -->
        <consonant ph="x" vc="-" ctype="f" cplace="v" cvox="-"/>

    <!-- d /d/ voiced alveolar stop/plosive -->
        <consonant ph="d" vc="-" ctype="s" cplace="a" cvox="+"/>

</allophones>
```

Developing a Part of Speech Tagger and a Corpus of Training Sentences for the Welsh Language

GRUFFUDD PRYS

Language Technologies Unit, Bangor University, Wales

GARETH WATKINS

Language Technologies Unit, Bangor University, Wales

In this chapter we present the early work done in creating TagTeg, a new Welsh part of speech (POS) tagger created using the spaCy software library. Such a resource is one of the most important and fundamental components of any natural language processing (NLP) pipeline. We also introduce the corpus of tagged sentences used to train TagTeg, which is a valuable and important resource in its own right. The chapter explores the technical and linguistic considerations that were taken into account when creating TagTeg and the corpus of tagged sentences, before proceeding to evaluate TagTeg's performance and reflect on the results. As this chapter presents our work at an early stage, we touch on our plans for the next steps in TagTeg's development.

Keywords: tagger, corpus, language technologies, Welsh, natural language processing (NLP)

1 INTRODUCTION

In this chapter we describe the initial version of a part of speech (POS) tagger produced as one of the natural language processing (NLP) components developed during the first 7 months of a resource development project. The project, *Text, Speech and Translation Technologies for the Welsh Language 2020 – 2021*, was undertaken by the Language Technologies Unit at Bangor University and was funded by the Welsh Government. It should be noted that at the time of writing we are at an early stage of the project, and although our progress has been considerable, more work is yet to be done. The purpose of this chapter is to report on our early progress and to take the opportunity to share our experiences and take advantage of peer feedback at the September 2020 Welsh Language Technologies Symposium.

In Section 2, we present the work on a statistical Welsh-language POS tagger that will form part of a natural language processing pipeline incorporating neural network methods. We provide an introduction to POS taggers and why we have created a new Welsh-language tagger. We discuss and compare two types of taggers: rule-based taggers, and statistical taggers. In Section 3 we describe the process used to create our new tagger and discuss the technology used. We go on to describe the corpus of sentences collected and annotated with POS tags in order to train the tagger. We describe the tagging process, the tag set, and some of the individual linguistic considerations. We then go on to report our initial results, before describing the next steps in the development of the tool. Finally, we summarize our conclusions in Section 4.

2 TAGTEG: A STATISTICAL WELSH LANGUAGE PART OF SPEECH TAGGER

In this section, we describe TagTeg, a statistical POS tagger created using the spaCy software library.[1]

2.1 What is a Part of Speech Tagger?

A POS tagger is a software tool that identifies the word class of a word in a piece of text. For example, in a short text such as 'big cat', a tagger would be expected to mark 'big' as an adjective and 'cat' as a noun, for example:

BIG/ADJECTIVE CAT/NOUN

Some of these word classes, such as noun, adjective, and verb, are familiar to most of us. However, some word classes that may be used by some taggers, such as the determiner word class, may be less familiar, and these may not necessarily be recognized by all taggers or linguistic frameworks. The boundary between some of these word classes can vary from one tagger to another. For example, one tagger might not recognize the determiner class, and will therefore tag some determiners as adjectives and others as pronouns. Assuming that the tagger is accurate, the tag used for such elements depends on the specific guidelines which have been set for that particular tagger to follow, as well the set of tags used by a particular tagger. This makes it difficult to compare the accuracy of different taggers to each other as they may follow different guidelines and use different tag sets (for more on this see [1]).

2.2 Justification for a New Part of Speech Tagger

In the case of Welsh, there are a number of POS taggers that are already available. One early example was developed to assist in tagging the first million word electronic Welsh corpus, CEG [2]. Another is the POS tagging module in the latest version of Cysill which replaced the hardcoded tagging rules of earlier versions [3]. In 2015, Cysill's tagger was made available as a free API service [4]. In 2010, the Autoglosser tagger was developed by Kevin Donnelly to tag Bangor University's multilingual Siarad corpus [5] of spoken texts. By 2018 a new version, called Autoglosser2 had been released [6]. Autoglosser2 focused only on Welsh. Eurfa [7], the lexicon which provided the lexical basis for Autoglosser, has also been used by other teams as the lexical basis for their POS taggers. These include WNLT [8] and its successor WNLT2 [9], both created by a team from the University of South Wales (funded by the Welsh Government), and CyTag [10], a tagger developed by a team from Cardiff University as part of the CorCenCC project [11] which was funded by a AHRC and ESRC grant.

A common feature of these taggers is that they are all rule-based taggers. However, according to the ACL's list of state-of-the-art English-language taggers, statistical taggers are now the best performing taggers in the field [12]. The top 8 taggers in the list report an accuracy of over 97% on the standard test set (which is a specific subset of the Penn Treebank texts originally taken from the American newspaper The Wall Street Journal). Some rule-based taggers have also reported results in the high nineties. However, such high scores can be a result of having developed the grammar rules on the evaluation data. The scores for statistical taggers tend to be more representative of their actual tagging performance when tagging text that was not encountered during the development process, where syntactic structures not found in the evaluation data are encountered.

For example, while Neale et al. [10] report that the CyTag tagger reached 96% accuracy in tagging data that was available during the tagger's development, our experiments have shown that its performance with texts which were not available during the development process is significantly lower [1] and that figures of around 82% are more representative of its performance on unseen texts. It was therefore our belief that a statistical tagger could provide

[1]www.spacy.io

accuracies at least 10% higher than such results, and would also be significantly better at recognizing the variety of names of people, places, organizations and products that arise within contemporary texts but which are difficult to account for comprehensively using vocabulary lists and gazetteers.

2.3 Other Work in the Field

Since starting work on TagTeg we have discovered references to work on Welsh language statistical taggers by Heinecke (based on UDPIPE) [13] and by Ezeani et al. [14] who have used word embeddings to train their Welsh language tagger. An evaluation of the Heinecke system was not yet available. Ezeani et al. have published good results, but they do not appear to be results from testing the tagger on unseen testing data. We were unable to test these systems on independent testing data so could not compare their system with TagTeg, as was done in [1] with the CyTag and WNLT2 systems. These systems appear to be work in progress, not yet been packaged and distributed for general practical use.

2.4 Rules Based Taggers and Statistical Taggers

2.4.1 Rule-Based Taggers

Broadly, rule-based taggers determine a word's POS by searching for the word in a comprehensive list of wordforms that have been associated with their corresponding parts of speech. For example, for an unambiguous wordform such as *lleol* (local), there should only be one POS associated with it in the list: adjective. As a result, tagging *lleol* appropriately is straightforward and no rules are required. Rules are used when there is more than one possible POS for a specific wordform. In such cases a rule is required to select the appropriate POS tag for that wordform based on the particular context in which it appears.

For example, the wordform *ceir* could represent a noun where it refers to more than one car, or it could represent a present impersonal form of the verb *cael* (to have). One way to distinguish between the two is by looking at the context of the surrounding words. For example, if *ceir* is preceded by the definitive article *'r*, it is likely that *ceir* is a noun meaning *more than one car* rather than a verbal form as *'r* would not be expected directly in front of an impersonal verb. This can be formulated as a rule, and by forming many such rules a tagger's ability to select the appropriate tag for different words sharing the same wordform can be improved. This is very important in Welsh, as many of the most common wordforms can represent a number of different words, including high frequency function words. For example, *y* can represent the English definite article *the*, but it can also represent a pre-verbal particle as seen in *yr aderyn **y** gwelsom ei nyth* (the bird we saw its nest). The way in which mutations are realized orthographically in Welsh can also lead to conflicting wordforms, as in the case of *nos* which can represent a mutated form of *dos*, as in *dyma fy **nos** cyntaf o'r feddyginiaeth* (this is my first dose of medicine), as well as its more common homograph which corresponds in English to *night*. With such complexity in the language, the problem with rule-based taggers is that experienced linguists must carefully stack many different rules atop each other so as to deal with all the grammatical variations of the language whilst also ensuring that these rules do not interfere with one another. Past a certain point, improving the accuracy of the tagger by adding more rules becomes increasingly unfeasible.

2.4.2 Statistical Taggers

Statistical taggers differ from rule-based taggers in that they do not rely on rules as their primary means of tagging. Rather, they operate primarily based on probabilities, that is, the probability that a wordform belongs to a particular

POS. In the case of a wordform such as *ceir*, a statistical tagger operates by determining the probability that *ceir* represents a form of the noun *car* versus the probability of it being a verbal form of *cael*, based on the context in which it occurs within the sentence. Statistical taggers learn a model of this probability by being trained on texts where the POS of each word has been annotated manually. Although annotating words with their POSs manually is not a simple task, it tends to be easier than trying to create, balance, and maintain hundreds of complex grammatical rules. Updating the tagger to cope with problematic texts is also easier as it is simply a matter of annotating and adding new example sentences. In practice, this has proven easier than trying to pile up increasingly complex grammatical rules to cope with exceptions to the grammatical rules of the language. Another benefit of statistical taggers is that they can learn to generalize what they have learnt during the training process so that they are able to tag wordforms that are unfamiliar to them. They do so based on a combination of features such as the position of a word in relation to other words, its prefix, suffix and capitalization patterns, and so on.

The main obstacle for statistical taggers is the amount of data that needs to be annotated before a useful tagger can be trained. Neale et al. [10] point to this as being one of the main reasons for choosing to develop a rule-based tagger rather than a statistical tagger. In order to create a reliable tagger that can cope with a variety of different types of Welsh (including different registers, dialects and topics), a large number of diverse texts must be annotated. For the tagger to achieve the best possible results, this training text must also include sufficient examples of the type of language the tagger intends to tag.

The copyright of the data used to train must also be considered. If the intention is to release and distribute the language model, consideration has to be given to whether a developer has the right to use the data to do so. This is discussed in detail by de Castilho et al. [15]. Perhaps most surprising is their view that models do not necessarily deserve their own copyright as they are not considered to be creative works in their own right, and therefore do not count as derivative works. However, the consensus in respect of language models, at least according to spaCy, seems to be that the model's licence copyright follows the data [16]. Despite the need to collect large amounts of data and ensure it is properly licenced, we believe that the advantages of the statistical approach of using data-trained models outweigh the disadvantages.

Another benefit of statistical taggers, in addition to their improved accuracy, is that many different taggers can be trained using the same training data. As the data exists independently of one particular tagger, the work of developing the data is not wasted when better statistical taggers are constructed (as is often the case with rule-based taggers when better taggers appear). Also, an approach focusing on the development of generalized tagged data, rather than on rules specific to one tagger, avoids restricting the investment in methods for tagging Welsh too closely to one tagger. There already exist a number of natural language processing libraries that could be trained on our tagged data (with some light reformatting of the data), including the John Snow Labs,[2] spaCy and UDPipe[3] libraries.

3 CREATING THE TAGGER

We chose the spaCy tagger as the basis for creating the TagTeg tagger. SpaCy is a natural language processing library. It includes a pipeline of language processing components or 'pipes', including a tokenizer, lemmatizer,

[2]https://www.johnsnowlabs.com/

[3]https://cran.r-project.org/web/packages/udpipe/vignettes/udpipe-annotation.html

named entity recognizer and a dependency parser, as well as the POS tagger. We chose to base our work on spaCy for a number of reasons. First, spaCy appears to provide a good compromise between accuracy and speed [17]. Some other libraries were a few percent more accurate, but that slightly higher accuracy tended to come at the expense of significantly higher processing time. Other considerations that contributed to the decision were the standard of spaCy documentation and support, and the fact that it was written in Python (which is a relatively simple coding language compared to other languages and easy to read and understand) in the more exposed layers of code. In addition, spaCy is distributed free of charge and under MIT's permissive open licence [18]. This gave us confidence that the work would be viable and sustainable. Another benefit of spaCy was its emphasis on providing a practical industrial standard for language processing. We believe that this fitted well with the desire of our Welsh Government funders to provide practical tools to support the use of Welsh in digital settings such as chatbots. SpaCy is used to provide the language-processing infrastructure for the Rasa [4] chatbot software and Microsoft's anonymization software Presidio,[5] for example. By working towards creating Welsh language provision for spaCy we hoped that general software developers would be able to easily add Welsh to the software stack they already intend to use, and that it can be used easily alongside other languages, including English.

3.1 spaCy Pipeline

As mentioned above, another motivation for developing a new Welsh tagger was the need to create a component that would neatly take its place as a single 'processing pipe' within the extended pipeline of language processing components contained within spaCy. These components may be interdependent. For example, by POS tagging a wordform such as *ceir* (which may represent the plural form of the noun *car* or a verb form of *cael*) the lexicon and POS can be passed to a lemmatizing component and properly lemmatized into either *car* or *cael* (have) (something WNLT2, for example, does not try to do). Tagging the POS of the words provides more information that may be used by further pipeline components, such as a named entity recognizer (for identifying entities such as names of people, organizations and places), and dependency parsers (for converting the text from a string of words to a syntactic tree showing the relationship of words to each other). It is useful for understanding the function of a word in a sentence, for example to establish what 'yn ateb' represents as it may correspond to the predicative 'yn' followed by a noun (e.g. *Mae B yn ateb anghywir* – 'B is the wrong answer'), or to a verbal particle followed by a verbnoun (e.e. *Dafydd oedd yn ateb y ffôn* – 'Dafydd was answering the phone'). These components provide important information for further NLP processes such as intent parsing systems, chatbots, text summarization systems and so on.

3.2 Data

3.2.1 Data Considerations

We have already emphasized the importance of data to the paradigm of training statistical language models. Having enough data to ensure that the models will be of a high standard is not merely a case of having access to data of sufficient size and variety; it must also be legally appropriate to distribute. In our case, we wanted to be able to distribute not only the trained models but also the training data itself. This would enable others to add their own

[4] https://rasa.com/

[5] https://github.com/microsoft/presidio

data to it, and train more tailored and larger models for their own purposes. It was also our intention to release the data under as permissive a licence as possible. Licences such as CC-BY-SA [19] and the various GPL [20] licences allow redistribution of the data but impose restrictions such as the need to provide attribution, or to release any derivative work under the same licence. On the other hand, permissive open licences such as CC0 [21] and MIT do not present such a barrier to the use of the data. These have proven to be more acceptable for industry use where the data can be used to create a commercial product without significant licensing limitations. As our aim was to provide resources that would make it easy to include Welsh in new products, this was a crucial consideration. We therefore decided to collect data that we were able to release under the CC0 licence, a license which is broadly equivalent the data having been place in the public domain. Where possible, we combined our efforts with the task of collecting sentences to contribute to Mozilla's Common Voice project,[6] where they would be used as recording prompts for recording volunteers speaking Welsh sentences (the aim being to train Welsh-language speech recognition models). Because of this, and our intuition that potential contributors would be more likely to contribute their data in the form of single mixed sentences rather than whole documents, this collection of texts is best viewed as a corpus of individual sentences rather than a corpus of documents and complete conversations (as is more true of the CorCenCC corpus, for example). Our CC0 corpus is therefore not intended to be a balanced corpus in the traditional manner. Rather, the aim is to ensure that it contains enough examples of the features that a model needs for training. As well as a selection of sentences from the Common Voice corpus, the data includes general sentences authored by members of the Language Technologies Unit at Bangor University, online chats, and translations of older English books and stories where the copyright had expired. In addition, there are also journal articles, tweets and encyclopaedia articles provided to us by their authors under the CC0 licence.

We have distributed the file for the first version in the jsonl file format, with one json entry for each line in the file, with the sentence text located within the "text" field, and information about it (including its original source, licensing details and its id within our system) in the metadata field. In addition to 3,345 complete Welsh sentences, the training set also includes 76,097 single word tagged 'sentences' taken from the Language Technologies Unit's lexicon which also forms part of Cysill and Cysgeir's word list. We used our extensive collection of in-house corpora to identify the wordforms. These one-word sentences have been included to boost the model's vocabulary, and the model is 1% more accurate as a result. Due to the nature of language, and the fact that the most common words are much more common than uncommon words, this means that in practice the model shows a marked improvement in trying to identify unusual words compared to a model not boosted in this way. Since the metadata of each sentence is attached to its jsonl record, and the source identified there, it is possible to exclude these one-word sentences from the data if required.

[6]https://commonvoice.mozilla.org/cy

3.3 Tagging

3.3.1 Tagging Tools

Prodigy,[7] the commercial package created by the software company Explosion[8] (spaCy's authors), was used to POS tag the hand-collected sentences. It was chosen because it provided a quick and easy way of POS tagging the words of the sentences in a way that was compatible with creating spaCy models.

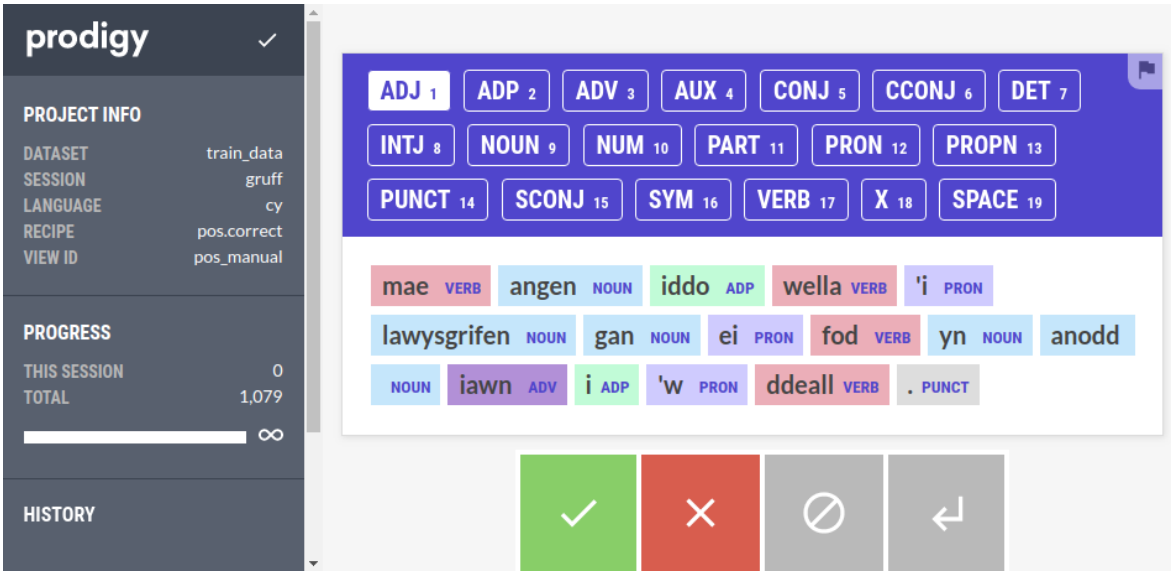

Figure 1: An example of the Prodigy tagging interface

3.3.2 The Tagset

The Universal Dependencies[9] tagset was used as it is a simple and familiar translingual tagset that would enable the syntax of Welsh to be compared with that of other languages, including in particular English, as both languages are used side-by-side in Wales.

3.3.3 Reconciling the Tagging

We sought to harmonize our use of POS tags with those found in the Bangor Welsh language lexicon.[10] This lexicon is a comprehensive list of Welsh language wordforms which was being prepared for release under the CC0 licence at the time. We also attempted to ensure consistency between the two human taggers responsible for the manual tagging of the data. This was achieved by having both individuals tag a large number of the same sentences and identifying where they had tagged wordforms differently. As a result, guidelines for the appropriate tagging of those

[7]https://prodi.gy/

[8]https://explosion.ai/

[9]https://universaldependencies.org/

[10]https://github.com/techiaith/lecsicon-cymraeg-bangor

instances were subsequently agreed upon, and the inconsistent sentences were re-tagged to ensure that there was consistency between them. At this early point in the project we have not yet established a score for inter-annotator agreement between different annotators. We intend to revisit this later in the project.

3.3.4 Linguistic Considerations

Discussions often arose surrounding how best to ensure consistency with regards to linguistic issues where no clear and unanimous resolution was available in the literature. In such cases, where we had to choose one method over another, pragmatic, practical solutions tended to be favoured.

3.3.4.1 Tagging verbnouns

One example of such a complex linguistic issue was the decision to treat verbs as nouns when they appeared to behave as nominals (such as *canu* in *y* ***canu*** *da*) and to treat them as verbs when they appeared to be more verbal in their appearance (such as *canu* in *yn* ***canu*** *yn dda*). This approach differs from that used by Heinecke [13] in his work on Universal Dependencies when developing their *Corpws Cystrawennol y Gymraeg*. As there is no specific top-level tag for verbnouns within the translingual set used by Universal Dependencies (see https://universaldependencies.org/u/pos/index.html for a list of these), a choice must be made from amongst the other available tags. Heinecke's practice with verbnouns is to tag every verbnoun as a noun. However, we felt that our method was closer to the current academic consensus, more closely aligned with the expectations of ordinary users, and would be useful to facilitate the distinction between actions and named objects. We hope to elaborate on this decision in the near future, and increase our dialogue and collaboration with Heinecke and other Universal Dependencies contributors.

3.3.4.2 Side effects of tagging the verbnoun

One of the side effects of the decision to tag verbnouns as verbs when their behaviour is more verbal than nominal is that it affects the appropriate method of tagging verbnouns when they form the individual elements of periphrastic constructions. When tagging verbnouns as verbs, we also decided to tag auxiliary verbs as AUX (auxillary) when they were found in periphrastic constructions. In doing so, we join the analyses of other languages within Universal Dependencies in recognizing auxiliary verbs. Auxiliary verbs are also recognized in the context of the Welsh language by scholars from the Welsh tradition such as Thomas [22]. Another related difficult decision was deciding on the appropriate tag for words such as *yn* and *wedi*, which were originally likely to have been prepositions [23] but now possess a time-aspect-mood (TAM) function. As these obviously perform a different function to the preposition, our initial response was to treat them as PART (particles), but in light of Universal Dependencies guidelines on TAM identifiers, this may need to be reconsidered in the future in favour of marking them as AUX. This change would however cause words such as *yn* and *wedi* to share the same tag as auxiliary verbs. The decision to accept or reject the possibility of tagging a verbnoun as a verb may therefore have implications for components that arise later in the pipeline, such as the dependency parser, and that the syntactic trees generated for the same sentence based on the two different interpretations may be very different. This is therefore an important consideration, and merits further research.

3.3.4.3 Tagging DET (determiners)

Another example where no clear precedent was available was when trying to implement Universal Dependencies' use of the DET (determiner) tag in Welsh. A general definition of a determiner is a word that helps refer more specifically to something, for example *the cat, some cats, your cat*. Note that these are words that, outside the context

of a phrase or sentence, are traditionally treated as pronouns. As a lexical class, determiners are a relatively new linguistic category in the field of linguistics. The class of determiners is not recognized by most Welsh dictionaries, including Geiriadur Prifysgol Cymru, which tends to label the words that would fit into the determiner category as demonstrative adjectives, as in the case of *hwn* (this).[11] Thomas [22] does discuss determiners to a certain extent, but we would like a more detailed discussion of them for our practical use.

Heinecke only uses the DET tag to mark the definite article in his work on Universal Dependencies. We feel, however, that the wider use of DET arose for practical pragmatic reasons, specifically to distinguish between forms of independent pronouns such as the *rhai* in *mae hi'n hoffi* ***rhai*** (she likes some) and the *rhai* in *mae hi'n hoffi* ***rhai*** *pobl* (she likes some people) where *rhai* provides more specific or detailed reference for a noun. We have chosen to follow this broader practice as we feel that the ability to distinguish between dependent and independent pronouns aids in correctly identifying the number of different elements in a clause as *rhai* in ***rhai*** *pobl* (some people) does not refer to an additional party but merely qualifies *pobl (*people*)*. Having said that, we recognise that the relationship between pronouns is complex and we believe that this will require further attention, particularly in terms of confirming the appropriate approach to reflexive pronouns e.g. ***dy*** *gath* ***di*** (your cat you).

3.4 Initial Results

3.4.1 Quantitative Evaluation

With the data available to us early in the project we have already succeeded in training an initial tagger with an accuracy of over 91% when evaluated on randomly collected unseen text that was not included in the data used to train the model. Although an initial result of 91% is very positive, and appears from our early experiments to be significantly more accurate than the Welsh language rule-based taggers evaluated [1], the results also suggest that there is room for improvement. In discussing English, Jurafsky and Martin [24] ascertain that a figure of 97% accuracy is the maximum that a human tagger could be expected to reach, for example. Assuming 500 words to one page, an accuracy of 91% would still result in approximately 45 incorrectly tagged words per page, with the errors potentially having significant implications for further downstream language processing tasks that are dependent on correct tagging. Therefore, whilst accuracy in the 90s may appear satisfactory, it is important to understand the standard of tagging represented by such scores, especially in the practical context of the applied use of the tagger.

Figure 2: An example of TagTeg output

3.4.2 Qualitative Evaluation

As these are the early stages of the development process, we must be content with providing a qualitative analysis rather than a complete quantitative analysis. A more general evaluation in comparison with other taggers will appear [1], and we will publish more detailed statistics regarding the tagger's performance later in the project. In the meantime, we present here some of our main impressions of the tagger's strengths and weaknesses so far.

[11]https://geiriadur.ac.uk/gpc/hwn.html

One of the most obvious flaws of the tagger currently is that it tends to tag every verbnoun as a verb, including those verbnouns that should be tagged as nouns. This is likely a result of having used one-word sentences to boost the model, as all verbnouns in the list were tagged as verbs. The examples of nominal verbnouns found within the complete sentences do not appear to occur in sufficient numbers in the data to correct the bias caused by this in the resulting model. One possible solution would be to add tagged trigrams such as *y canu da* and *yn canu yn* rather than tagged individual words, and we intend to put this method to the test. We also doubt that the proportion of nominal verbnouns found in ordinary Welsh sentences would be high enough to adequately influence the model, so we may need to be reinforce this aspect of Welsh syntax within the training data, or at the very least within the development data (the data used to evaluate the model's gain in accuracy as it is being trained). We expect that a tagger that tries to distinguish between a nominal verbnoun such as *canu* in *y **canu** da* and a verbal verbnoun such as *canu* in *yn **canu** yn dda* is likely to score lower than a tagger that treats all verbnouns as belonging to the same lexical class. This is simply because such a tagger would not have to make a decision in such cases and therefore cannot make an incorrect classification. We would, however, suggest that the results of a tagger which does not distinguish between the verbal and nominal roles of verbnouns is not as useful as one that does in many applied contexts.

Another, somewhat unexpected shortcoming of the tagger is that it will sometimes identify personal names (such as *Sioned*) as verbs. We believe that this may be the influence of verbal forms such as *sonied* and the general influence of verbs ending in -ed on the model. We hope to see this trend disappear as we tag more and more complete sentences and reduce our reliance on 'single word sentences'.

3.5 Using the Tagger

An initial version of TagTeg is available to download from https://github.com/techiaith/model-tagiwr-spacy-cy. Anyone can use the data to train their own spaCy 2 model by using spaCy's convert command on the command line to convert the file to the current training format (see https://spacy.io/api/cli#convert), and then use the spaCy train command to train the model. To use the tagger, see the general spaCy documentation.

3.6 Further Work

As noted in section 1, there remains further work to be done on TagTeg. Our intention is to collect and tag additional data in order to not only increase the number of sentences in the corpus of tagged sentences but also to improve the variety found within the data. We anticipate that doing this, once we retrain the model on the enlarged corpus, will improve TagTeg's output. As noted in section 3.4.2, we will conduct a full evaluation of TagTeg, including an evaluation using the accuracy and recall metric and the F1 metric. In addition, we aim to evaluate the inter-annotator agreement between human annotators to ensure consistency between them, and then verify and reconcile any differences in annotation. We also plan to present our work to other academics, and engage in discussions with the international community responsible for the initiation and oversight of Universal Dependencies.

4 CONCLUSIONS

In this chapter we have discussed the early progress made during the creation of TagTeg, a statistical Welsh language POS tagger. Having explained exactly what a POS tagger is, we discussed recent and historical developments in the field, and detail two different types of tagger. We went on to describe the process of creating

TagTeg, including a description of the language and copyright considerations, and gave a brief description of the technology used in tackling the task. We have also summarized the findings of the evaluation of TagTeg to date.

It is still early days, and there is more work to be done to improve the tagger and provide a deeper, more meaningful evaluation framework. However, in light of the results of the evaluation so far, we are encouraged by TagTeg's ability to tag text efficiently and accurately. As it stands, this is an important contribution to the Welsh language's digital toolset and its ability to provide solutions to some of the technical challenges of today's world.

ACKNOWLEDGMENTS

This work was funded by the Welsh Government as part of the Text, Speech and Translation Technologies for the Welsh Language 2020 – 2021 project.

References

[1] Gruffudd Prys and Gareth Watkins. 2021. Gwerthusiad o Dri Tagiwr Rhannau Ymadrodd Cymraeg. Arxiv.org (forthcoming).

[2] Nick C. Ellis, C. O'Dochartaigh, William Hicks, M. Morgan, and N. Laporte. 2001. Cronfa Electroneg o Gymraeg (CEG): A 1 million-word lexical database and frequency count for Welsh. [On-line]

[3] William Hicks. 2004. Welsh Proofing Tools: Making a Little NLP go a Long Way. In Proceeding of the1st Workshop on International Proofing Tools and Language Technologies. Univeristy of Patras, Greece.

[4] Uned Technolegau Iaith Canolfan Bedwyr Prifysgol Bangor. 2015. API Tagiwr Rhannau Ymadrodd. Retrieved March 3, 2021 from https://github.com/PorthTechnolegauIaith/postagger/blob/master/README.md

[5] Kevin Donnelly and Margaret Deuchar. 2011. The Bangor Autoglosser: A Multilingual Tagger for Conversational Text. In Proceedings of the Fourth International Conference on Internet Technologies and Applications (ITA 11). Glyndwr University, Wrexham, North Wales, 17-25.

[6] Kevin Donnelly. 2018. Autoglosser2 released. Retrieved September 1, 2020 from http://kevindonnelly.org.uk/2018/02/autoglosser2-released/

[7] Kevin Donnelly. 2013. Eurfa. Retrieved February 24, 2021 from http://eurfa.org.uk

[8] Daniel Cunliffe, Doug Tudhope and Andreas Vlachidis. 2016. The Welsh Natural Language Toolkit WNLT & CYMRIE. In WNLT Final Workshop. University of South Wales, Trefforest, Wales.

[9] Daniel Williams. 2017. Welsh Natural Language Toolkit. In WNLT Final Workshop. University of South Wales, Trefforest, Wales.

[10] Steve Neale, Kevin Donnelly, Gareth Watkins and Dawn Knight. 2018. Leveraging Lexical Resources and Constraint Grammar for Rule-Based Part-of-Speech Tagging in Welsh. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). LREC, Miyazaki, Japan, 3946-3954.

[11] Dawn Knight, Steve Morris, Tess Fitzpatrick, Paul Rayson, Irena Spasić and Enlli-Mon Thomas. 2020. The National Corpus of Contemporary Welsh: Project Report | Y Corpws Cenedlaethol Cymraeg Cyfoes: Adroddiad y Prosiect. Cardiff University, Wales. arXiv:2010.05542.

[12] Association for Computational Linguistics. 2020. POS Tagging (State of the art). Retrieved September 1, 2020 from https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

[13] Johannes Heinecke and Francis M Tyers. 2019. Development of a Universal Dependencies treebank for Welsh. In Proceedings of the Celtic Language Technology Workshop. European Association for Machine Translation, Dublin, Ireland, 21-31.

[14] Ignatius Ezeani, Scott Piao, Steven Neale, Paul Rayson and Dawn Knight. 2019. Leveraging Pre-Trained Embeddings for Welsh Taggers. In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019): Held at the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy. Association for Computational Linguistics, Florence, Italy, 270-280.

[15] Richard Eckart de Castilho, Giulia Dore, Thomas Margoni, Penny Labropoulou and Iryna Gurevych. 2018. A Legal Perspective on Training Models for Natural Language Processing. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). LREC, Miyazaki, Japan, 1267-1274.

[16] Adriane Boyd. 2021. In which license falls a model trained using spacy's LGPL models ? . Retrieved March 3, 2021 from https://github.com/explosion/spaCy/issues/7216#issuecomment-787816212

[17] Jinho D. Choi, Joel Tetreault, and Amanda Stent. 2016. It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Beijing, China, 387-396.

[18] Fossa. 2021. Open Source Software Licenses 101: The MIT License. Retrieved March 3, 2021 from https://fossa.com/blog/open-source-licenses-101-mit-license/

[19] Creative Commons. 2021. Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). Retrieved March 3, 2021 from https://creativecommons.org/licenses/by-sa/4.0/

[20] GNU. 2021. GNU General Public License. Retrieved March 3, 2021 from https://www.gnu.org/licenses/gpl-3.0.html

[21] Creative Commons. 2021. CC0 1.0 Universal (CC0 1.0) Public Domain Dedication. Retrieved March 3, 2021 from https://creativecommons.org/publicdomain/zero/1.0/

[22] Peter Wynn Thomas. 1995. Gramadeg y Gymraeg. Gwasg Prifysgol Cymru. Caerdydd.

[23] Patrick Sims-Williams. 2015. The four types of Welsh YN. Transactions of the Philological Society. 133, 3 (November 2015), 286-304. https://doi.org/10.1111/1467-968X.12052

[24] Daniel Jurafsky and James H. Martin. 2020. Speech and Language Processing (3rd ed. draft). Retrieved March, 2021 from https://web.stanford.edu/~jurafsky/slp3/ed3book_dec302020.pdf

Welsh Word2vec model: vector representation of the semantic correlation of Welsh words based on their embeddings within an enormous Welsh corpus

GRUFFUDD PRYS

Language Technologies Unit, Bangor University, Wales

GARETH WATKINS

Language Technologies Unit, Bangor University, Wales

We report on an early output from the Text, Speech and Translation Technologies for the Welsh Language 2020 - 2021 project: a large collection of word2vec word vectors, trained on a substantial corpus of Welsh, which models the semantic interrelation of Welsh words so as to facilitate language processing tasks. We describe its creation and discuss its usefulness for language processing, and provide an initial qualitative evaluation of the model itself.

Keywords and Phrases: Welsh, word2vec, corpus, word vectors, word embeddings, language technologies, vectors

1 INTRODUCTION

This chapter discusses a sizeable Welsh word2vec model which is useful for many different NLP processes, and serves as a useful tool for lexicographical and terminological research. In Section 2, we introduce embeddings, word vectors and word2vec models in a manner intended for an audience unfamiliar with such concepts. In Section 3 we describe the textual data used during training as well as our attempts to collect a substantial amount of Welsh texts that would be comprehensive in nature and represent a wide variety of Welsh registers, styles and subject areas. In Section 4 we describe the technical details of the training process. We proceed in Section 5 to describe the results of the training process and its outputs. In Section 6 we present a qualitative evaluation of the model. We seek to highlight some of the weaknesses and virtues of such models, and discuss other relevant considerations, focusing especially on those considerations that are especially relevant to the Welsh language. Finally, in Section 7, we summarize our findings and conclude by outlining the main points raised in this chapter and by noting what we consider to be the next steps in terms of priority so that the work can be developed further.

2 AN INTRODUCTION TO EMBEDDINGS, WORD VECTORS AND WORD2VEC

Word2Vec [1] is a method for representing the relationship between wordforms using numeric vectors. The aim is to create a numerical representation of a wordform based on its relationship with the other wordforms which occur around it. This 'embedding' approximates semantic information, and has proven to be an effective way of enriching textual data with information that in general aligns well with its semantic meaning.

One of the main advantages of a method such as word2vec is that it enables the mathematical manipulation of language. By representing the words using a numerical form, we can measure the similarity of different wordforms, and group together wordforms that share a similar meaning. This can be useful in lexicographical work, and since 2016 the Language Technologies Unit's terminologists have been using word2vec vectors based on the Cysill Online Corpus [2] to identify words and terms with similar meanings. Palmer and Spasic have also worked on Welsh language embeddings as part of the Welsh Government's Welsh words by numbers: "Wales" + "capital" = "Cardiff" project in 2019 [3].

One of the main advantages of a numerical representation of words is that it allows us to treat language on a more general basis. As well as assisting with lexicography, these embeddings can enable machine translation techniques to consider the use of semantically similarly words when translating [4], or enhance the ability of part of speech taggers to classify words appropriately into categories such as nouns, verbs , adjectives [5]. With the word2vec method, neural networks are used to train the word2vec model on large collections of text. The model generated by the training process takes the form of an extensive list of the words found in the training text (but excludes the most unusual words) alongside their corresponding vectors. The vector of each word represents the relationship of the wordform to all other wordforms found in the text of the training corpus. These vectors act as coordinates for the wordforms within the conceptual vector space, as if they were points on a map, with the wordforms that are most similar in their usage being grouped closer together within that conceptual space. These vectors take the form of a matrix within the word2vec model. In Figure 1, we show a vector for the Welsh wordform 'ci' (dog). It is these matrices which are used when operating mathematically on the vectors.

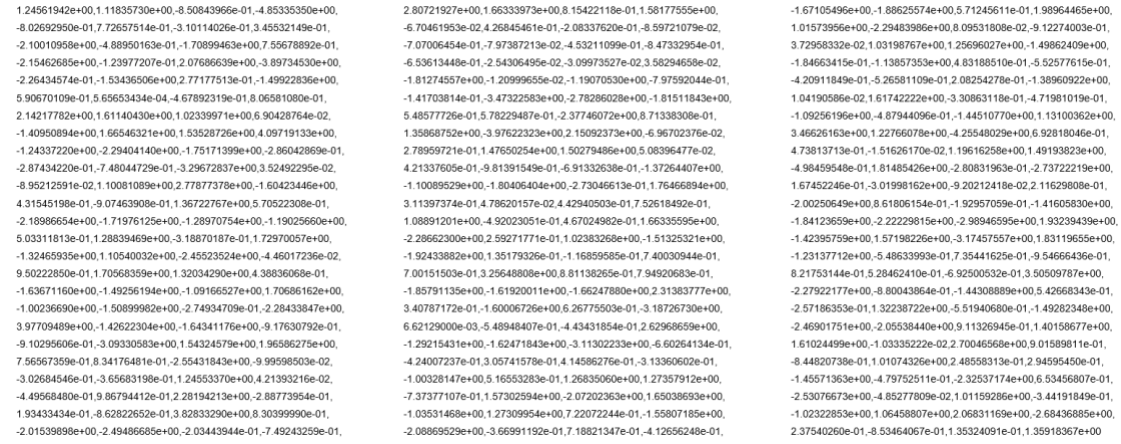

Figure 1: A matrix showing a vector for the Welsh wordform *ci* from our word2vec model (note that it is organized across three columns to save space)

A more manageable form for comparing the values of wordforms is the L2 norm of the matrix. For example, the L2 norm for the wordform *teacher* could be expressed to two decimal places as 0.63, which is easier us to use than the matrix in its entirety. We will use L2 norms when evaluating and comparing the similarities of wordforms in Section 6.

3 THE TRAINING CORPUS' DATA

To produce a word2vec model which provides vectors which are appropriate for the Welsh language in general, the training data used must include a wide variety of the different types of Welsh found in the textual form. This entails the collection of data from a wide variety of registers, styles, dialects, subject areas, and so on. This in turn requires that a large amount of data be collected, and for that data not to be of a repetitive nature.

3.1 Data Collection

Undoubtedly, collecting sufficient suitable data is the main challenge when training a high quality model of word embeddings, especially in the context of a language such as Welsh where comparatively fewer resources are

avaialable. For a number of years the Language Technologies Unit has attempted to gather together Welsh-language text corpora so as to be able to facilitate computerized language research. Some of the resources and services provided by the Unit to the wider community were designed specifically to function as a source of textual data that would enable the use of more recent NLP techniques that require large amounts of data to work. The Cysill Online service is an obvious example of such a service.

3.1.1 Cysill Online Corpus (334.9 Million Tokens)

From the outset, Cysill Online was developed not only to provide a useful service to those who write in Welsh, but also as a means of collecting a substantial corpus of Welsh text for the internal research purposes of the Language Technologies Unit [2]. This was achieved by providing a useful grammar checking service in return for the user's agreement for their data to be used for research purposes. In these texts we find representation for all kinds of Welsh as written today. However, we believe that there is a strong tendency in the data towards texts from the education context. This reflects the relative strength of the Welsh language in that aspect of life, and the fact that Cysill Online is a particularly useful tool for those who study and teach within that sector. The texts are largely non-edited. As a result they show varying ability in the standard of the Welsh used, and contain more spelling and grammar errors than are found in texts subjected to an editorial process. Unfortunately, as users of the service sometimes submit texts of a private or sensitive nature for validation, it is not possible for the data to be shared in a format where the texts could be read. However, since word2vec models are an approximate, generalized representation of the relationship of individual wordforms, the models can be shared without releasing sensitive information. This is an active monitor corpus, and although already large, it continues to increase in size as the Cysill Online service is used.

3.1.2 DECHE Corpus (2.1 Million Tokens)

The DECHE project [6] was another example of a project which aimed to create a corpus of texts hand in hand with the creation of a more general resource. This project was attractive to the funder, the Coleg Cymraeg Cenedlaethol (the federated Welsh-language university), as a means of assembling a ready-made digital library of historically Welsh scholarly resources in the form of e-books and PDF volumes. It was also a way of collecting a Welsh corpus of polished scholarly writing that included the work of many giants from the Welsh scholarly tradition such as J. R. Jones and Bobi Jones. This corpus can be searched at www.corpus.cymru.

3.1.3 Vocab Corpus (158.0 Million Tokens)

As part of an agreement with organizations using the Unit's Vocab service on their websites, an extensive corpus of texts belonging to those organizations was collected [7]. Most of these texts are collected from contemporary journalistic sources.

3.1.4 Corpus of Education (7.2 Million Tokens)

A corpus of educational materials compiled to facilitate terminology research by the Language Technologies Unit and in particular to aid with the standardization of Welsh-language terminology [8].

3.1.5 Corpus of Bangor University Materials (33.5 Million Tokens)

This corpus contains Welsh language materials produced by Bangor University and which was made available to us, including information from the University website. It mostly contains text related to university activities rather than content form specific subject fields.

3.1.6 Corpus of Tweets (8.2 Million Tokens)

This is a corpus of Welsh language tweets collected by the Language Technologies Unit in 2015 [9]. following a change in Twitter's terms of service we can no longer distribute the corpus itself. Those wishing to access Twitter's data must now download the data directly (at a limited rate) from Twitter.

3.1.7 Corpus of The Record of Proceedings (21.5 Million Tokens)

This is a corpus containing Welsh language sections of the Record of Proceedings of the various Welsh national 'Assemblies'. It includes the 1999-2003 Assembly and the 2007-2010 Assembly, which are both available from HMSO and the National Assembly and which hold Crown copyright. The data for those Assemblies can be searched here: http://corpws.cymru/ycofnod.

3.1.8 Gwerddon Corpus (0.7 Million Tokens)

This is a corpus drawn from the volumes of the Coleg Cymraeg Cenedlaethol's Gwerddon Higher Education academic journal. It includes academic discussions on a variety of subjects relating to the Higher Education fields taught at Welsh Universities. It provides an important technical vocabulary of those fields, as well as an important technical context for them.

3.1.9 Y Gwyddonydd Corpus (2.4 Million Tokens)

Y Gwyddonydd was a Welsh-language scientific magazine that ran from 1963 to 1996. We originally collected the texts for terminological research from files provided by the National Library of Wales as part of their Welsh Journals project. It provides important information about Welsh scientific vocabulary.

3.1.10 Corpus of Bibles (2.9 Million Tokens)

The Bible represents an important corpus for many languages [11]. This includes the Welsh language, for which several different versions exist [12]. We used a corpus constructed from the contents of three different versions of the Bible, namely the bible commonly known as the Beibl William Morgan (the William Morgan Bible, which is in fact an orthographically revised version of that which originally appeared in 1588), the Beibl Cymraeg Newydd (New Welsh Bible) published first in 1988, and the recent contemporary translation, beibl.net. Whilst including three versions of the Bible may mean that certain wordforms and structures could be over-represented in the data, we judged that a number of different reasons justified their inclusion.

Firstly, there is a great deal of lexical variation between the three versions. The Beibl William Morgan in particular contains concise verbal forms that are now less common in Welsh and would be difficult to find in more modern sources. One justification was that the inclusion of the Beibl William Morgan would mean that the trained model would be familiar with historically more frequent concise forms of the verb such as *cychwynasant* and *cychwynnant* (they started) as in the phrase *yn olaf y cychwynnant* (they started last, e.g. Numbers 2:31).

In comparison, Y Beibl Cymraeg Newydd represents a style that is modern in comparison, but which was still fairly conservative even when it was published in 1988. An example of this trend is that it uses *chwi* (you) for the second person plural pronoun, although CEG [12] shows that *chi* was much more commonly used during the same period (1375 instances of *chi* compared to 93 of *chwi*). The style Y Beibl Cymraeg Newydd is more periphrastic in its choice of verbs than that of Beibl William Morgan. To continue with the example from Numbers 2:31, we see constructions such as *fydd yr olaf i gychwyn allan* (will be the last to start out) used in the 1988 Bible, compared to *yn olaf y cychwynnant* in the William Morgan Bible.

By the time beibl.net was published, the syntax is even more periphrastic, and more regularly so. The passage found in this version of the Bible is *fydd yn symud allan olaf* (will be the last to move out). One notable difference between the three bibles was that the number of unique words reduced significantly from the earliest to the latest. We believe this to be a result of the syntax developing from the use of concise verbal forms to the use of increasingly periphrastic verbs where there is less variation in the individual wordforms used. Compare, for instance, *sylwasom* (we noticed) and *rydym wedi sylwi*, where *sylw*, *rydym* and *wedi* are less likely to be unique within the broader text than *sylwasom*).

Without including these earlier bibles it was felt that obtaining high-quality embeddings for concise forms of the verb would prove difficult using contemporary sources alone. It was also felt that this would result in the traditional syntax of the Bible not being reflected more generally in the embeddings, and that this would be a weakness given the influence of the Bible on the language. There are certainly disadvantages with including the Bible three times in the data, the most obvious being that certain forms found in the Bible are over-represented compared to the rate in which we would expect to see them in general Welsh texts. This aspect is discussed in more detail in Section 6.11. However, due to the variation in syntax from one version to another, we do not believe this to be overly problematic, especially considering that the Corpus of Bibles' size represents only a relatively small part of the size of the wider training corpus.

3.1.11 Encyclopedic Corpus (Wicipedia) (14.3 Million Tokens)

This corpus was created on the basis of an export of Wicipedia, the Welsh version of Wikipedia. It was judged important to include this corpus as the inclusion of a general encyclopedia of knowledge would act as a valuable source of general semantic information. However, significant effort was required to data, primarily due to many of the exported sentences being variations on the same template sentence. For example, the corpus contains hundreds of sentences on the pattern *The height of the summit from sea level is [number] meters ([number] ft)* or *The height was measured and confirmed on [date] [month] [year]*. Although the Wikipedia corpus seemed large prior to the deduplication process, the deletion of semi-duplicate sentences meant that the size of the final corpus was significantly reduced. However, without the de-duplication process, sentences like the above would have unduly influenced the trained model.

3.1.12 Massive Corpus of the Web (25.1 Million Tokens)

The most significant 'corpus' of Welsh today is likely the Welsh texts available on the web. We used a web crawler to specifically target and collect Welsh texts from the web to build a corpus of web texts,. This was done by collecting the URLs of known Welsh language websites to act as starting points for the scraping process, and by specifically excluding prominent websites which we knew did not contain Welsh text. Attempts were also made to exclude websites that included machine translation. The scraper only kept unique sentences, and this does have

implications for calculating the frequency of word usage. However, because there are so many template sentences on the web, we felt that keeping every single example of particularly common texts would have been more problematic than discarding the duplicates. We used cld2[1] on the output to filter out sentences in other languages that had been accidentally collected despite the scraper's focus on Welsh. We believe that the remaining sentences in the corpus are almost all valid Welsh. We hope to continue to increase the size of this corpus in further versions.

3.1.13 Corpora not included

For a variety of reasons, we chose not to include in our training corpus a number of corpora that we might have been expected to use. We chose not to include CEG so that it could be used as a testing and comparison corpus for future evaluation of the training corpus. The Siarad Corpus [13] was not included as the data would need to have been substantially reformatted before the text was useable. We did not include OSCAR [14] or any Welsh subdivisions of CommonCrawl[2] as we were concerned that this could lead to duplication in the data. CorCenCC [15] was not available to us when the model was trained.

3.2 Data Cleaning

For the best possible results it was important to be clean the data before training. This included the removal of duplicate sentences, especially ones arising from website interface templates and repeated copyright statements. Encoding can also be problematic, as websites do not always use the appropriate UTF-8 encoding for Welsh texts (which is essential for the proper conveyance of 'ŵ' and 'ŷ'. All of these issues demonstrate the importance of analyzing any training corpus to ensure that error or corruption in the data does not unduly affect the results.

3.3 Training Corpus Content

Below is a brief overview of the size of the total training corpus used to train our word2vec model, along with a breakdown of the individual corpora it contains. It should be emphasized that these are approximate figures as the final tokenization implementation was not complete when this initial model was trained (we intend to recreate a new model once the final tokenization has been determined within the extended pipeline).

Table 1: Contents of the Corpora

Corpus	Number of Tokens
Cysill Ar-lein Corpus	334.9M
Vocab Corpus	158.0M
Bangor University Materials Corpus	33.5M
Massive Web Corpus	25.1M
Record of Proceedings Corpus	21.5M
Encyclopaedic Corpus	14.3M
Twitter Corpus	8.2M
Education Corpus	7.2M
Corpus of Bibles	2.9M
Y Gwyddonydd Corpus	2.4M
DECHE Corpus	2.1M
Gwerddon Corpus	0.7M

[1]https://github.com/CLD2Owners/cld2

[2]https://commoncrawl.org/

Corpus	Number of Tokens
Cysill Ar-lein Corpus	334.9M
Total	**610.9M**

At 610 million tokens, this is the largest Welsh language research corpus that we are aware of (though it is likely that some of the largest software companies such as Google and Facebook may possess more Welsh data). We believe the corpus represents well the different types of Welsh produced and distributed on the web. However, it does not necessarily reflect a balanced representation of the use of Welsh in general as it was not our aim to create a formal balanced corpus of Welsh as was the intention of the CorCenCC project. The advantage of not aiming for such balance is that it facilitates gathering greater quantities of data. This is reflected in the fact that this training corpus is 610M in size while the CorCenCC corpus is 14M. This, to a certain extent, illustrates the tension between trying to gather as much data as possible and trying to collect a more selective and balanced corpus.

3.3.1 Redistribution of the Corpora

Note that that the copyright for the majority of the texts used to train the model is not held by Bangor University, and although we are able to release the model in the form of a list of individual wordforms and their vectors representing our research, we do not have the right to redistribute the text for the majority of these corpora where we are not the copyright holders. In certain cases, as with the Cysill Online Corpus, the benefits of distributing such substantial corpus publicly are also overridden by the need to protect the privacy of our users.

4 TRAINING

Since the introduction of word2vec, other, more sophisticated methods of creating word embeddings have emerged, including the Glove and FastText methods which seek to address some of word2vec's weaknesses. Nevertheless, we commenced our work by training a word2vec model as this was the word embeddings format supported by spaCy's language processing library at the time, and our work on word embeddings forms part of a broader attempt to develop a pipeline of Welsh language processing pipes based on said library.

The Gensim software package[3] was used to train the model. The process was relatively simple – certainly less challenging than the task of obtaining data of sufficient size and variety. Gensim also facilitates the training of Fasttext models [16], and we plan to do so in the near future to be able to compare our model with others such as the pretrained Fasttext model provided by Microsoft.[4] We would expect the results for a Fasttext model trained on the same data to improve upon on those for a word2vec model as Fastext enables embeddings to be created for subword units. This enables the method to provide a vector for wordforms not found in the model (Out of Vocabulary words or OOVs) based on a combination of subword elements that are found in the model (see Section 6.11 for a fuller discussion).

4.1 Training Settings

In training the model using Gensim, we chose to use a value of 300 dimensions for the vector dimensions as this is considered to be at the upper end of what is considered to be a normal value for training [17], and we could

[3]https://github.com/RaRe-Technologies/gensim

[4]https://github.com/microsoft/nlp-recipes/blob/master/utils_nlp/models/pretrained_embeddings/fasttext.py

therefore expect useful results without being hampered by a lack of dimensionality. The threshold for the number of times that a wordform had to occur in the corpus to be included in the model was set at 5. That seemed to us, after some experimentation, to be a good compromise between the inclusion of invalid wordforms and the exclusion of valid but rare wordforms.

As our intention with this initial work was to establish a baseline for further experimental research, we kept Gensim's default settings, only adjusting the default vector size from 100 to 300. As a result, the default CBOW method was used rather than the Skip-Gram method (see https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314), but we intend to experiment with this in the future.

5 RESULTS

The training resulted in the production of a model comprising a total of 292,769 unique wordforms together with their corresponding 300 dimension word vectors. We decided to filter out alphanumeric tokens such as *abc123* and tokens which were a mix of upper and lower-case such as *digwyddiadMawr* (eventBig), so the 292,769 wordforms do not include such tokens. The wordforms were not normalized in terms of their capitalization, so wordforms like *gwen* (=white) and *Gwen* (a person's name) are different entries which have different vectors within the data. These wordforms include common Welsh words, proper names (such as names of people, places and products etc.), as well as some common foreign language wordforms (mostly from English). The wordforms within the model are recorded in order of frequency in the corpus, which is a very useful feature for a variety of language processing tasks.

Table 2: Most common wordforms and their corresponding L2 Norm

The most frequent tokens in the training corpus (excluding punctuation but combining lower and upper case)			
1.	yn	11.	wedi
2.	i	12.	ei
3.	y	13.	am
4.	‘r	14.	bod
5.	o	15.	gan
6.	ar	16.	mewn
7.	mae	17.	eu
8.	‘n	18.	cael
9.	yr	19.	fod
10.	ac	20.	hyn

We provide the word2vec model trained on the basis of the training corpus in the form of a .bin file at https://github.com/techiaith/word2vec-en, along with instructions on how to use it. The file is 337Mb in size.

6 QUALITATIVE EVALUATION

Although quantitative methods of evaluating the quality of embeddings have been developed [18], we have not yet had the opportunity to prepare the necessary Welsh evaluation data. As was the case with the independent evaluation of these embeddings commissioned by the Welsh Government, we must turn instead to a qualitative evaluation of the data. The anonymous evaluator [19] gave the following feedback:

> The distribution contains a sample script for testing the vectors, including examples on testing several similarity scores for the cases 'teacher', 'school' and 'king' delivering excellent results.

We list the results for those wordforms below for discussion and evaluation.

6.1 An Evaluation of Athro, Disgybl, Coleg and Ysgol

Below we show the wordforms returned by the model after using Gensim's most_similar feature to find the most similar wordforms to both *athro* (teacher) and *disgybl* (pupil). The higher the similarity score, the more similar the listed wordform is to the evaluated wordform according to the model.

Table 3: Wordforms similar to *athro*

Wordforms similar to **athro**	Similarity (L2 Norm)
athrawes	0.8889025449752808
ymarferydd	0.737335205078125
addysgwr	0.7125141024589539
athrawon	0.6787909865379333
ymarferwr	0.6760769486427307
tiwtor	0.6219297051429749
hyfforddai	0.6186233758926392
cymhorthydd	0.5997982025146484
Athrawes	0.5851731300354004
ymchwilydd	0.5845403671264648

Table 4: Wordforms similar to *disavbl*

Wordforms similar to **disgybl**	Similarity (L2 Norm)
dysgwr	0.909702718257904
plentyn	0.7796590328216553
disgyblion	0.7423983812332153
dysgwyr	0.7294760346412659
cyfranogwr	0.710708737373352
myfyriwr	0.6880736351013184
unigolyn	0.6173608303070068
ddisgyblion	0.6116431355476379
plant	0.5878321528434753
cyfranogwyr	0.5865952968597412

As we see in table 3, the concepts listed are very relevant to the meaning of the word *athro*. In the list, the most similar concept to *athro* is *athrawes* (a female teacher). As gender is the only thing that separates *athro* and *athrawes* it is difficult to think of a more similar concept to athro, especially as the gender of the educator has no special significance within education. Many of the most similar wordforms are hyponyms or supercategories of *athro*. We see that a teacher is a kind of *addysgwr* (educator) as well as a kind of *ymarferydd* (practitioner) or *ymarferwr* (also practitioner). Some of the other results represent similar but slightly different concepts. A *tiwtor* (tutor) may, for example, represent an educator who teaches one to one, while a *cymhorthydd* (assistant) is usually a less qualified additional educator. There is a similar pattern in the results for *disgybl*, which are also relevant and useful.

One feature that emerges is the existence of several of the basic forms' inflected forms, including both plural and singular forms of the same word. We are presented with *athrawon* (teachers) as a plural form of teacher, for example. There is also the capitalized form of *Athrawes*, which in this case represents the same concept as *athrawes*, but capitalization may also refer to different concepts, such as in the case of *glesni* (=blue or greenness) and *Glesni* (female's name).

In the results for *disgybl*, there is also an example of the mutated form *ddisgyblion* (pupils). This highlights the fact that mutation of initial consonants, as seen in Celtic languages, have significant implications for the creation of word embeddings. This the information contained in the vectors is shared and spread between more different wordforms than would be the case if there were no mutations in the language. We therefore suggest that the information contained within mutated word embeddings is poorer and more fragmented (spread across multiple forms) than the language's nonmutable words, or the equivalent words of other less morphologically complex languages.

To turn to a specific example, the information for all the forms which represent the Welsh word for *pupil* is shared between *disgybl*, *nisgybl* and *ddisgybl* (not to mention the plural forms and mutated forms of those plurals). Although the results for *disgybl* in table 4 are good, we see that those for *nisgybl* in table 5 do not seem to be as good (those for *ddisgybl* in table 6 may be more relevant).

Table 5: Wordforms similar to *nisgybl*

Wordforms similar to **nisgybl**	Similarity (L2 Norm)
nghyfrifiannell	0.8409432768821716
nghanolwr	0.840394139289856
nghywaith	0.8396995067596436
nysgwr	0.8396143913269043
ngweddu	0.838886022567749
nghymhorthydd	0.8387882709503174
nghamddealltwriaeth	0.835710883140564
nghyfnitherod	0.8352251052856445
nghynorthwywyr	0.8346971273422241
ngwahaniaethu	0.8331178426742554

Table 6: Wordforms similar to *ddisgybl*

Wordforms similar to **ddisgybl**	Tebygrwydd (L2 Norm)
fyfyriwr	0.7124826908111572
ddysgwr	0.706892728805542
blentyn	0.6756373643875122
fachgen	0.674723744392395
berson	0.5963745713233948
gystadleuydd	0.5879442691802979
fyfyrwraig	0.5868666172027588
gyfranogwr	0.5710302591323853
gymhorthydd	0.5595928430557251
disgybl	0.5465054512023926

Note also that the wordforms found in the lists of words most similar to a word tend to share the same mutation as that word. In table 5, each result shares the same nasal mutation as the comparison word, namely *nisgybl*. We believe this to be the shadow of the specific precursor words which trigger the mutation in the specific mutated wordform. For example, the pronoun *fy* (my) and the conjunction *neu* (or) will always cause a mutable word that follows it to mutate softly [20]. Since embeddings operate on the principle that words are characterized by the company that they keep [21] (that is, the words that surround them), it is inevitable that the embeddings of mutated wordforms would be heavily influenced by the preceding words which triggers the wordform's mutation. However, it is surprising that that pattern is so clear in the *nisgybl* results. We feel that the emergence of nasal forms such as *nghyfrifiannell* (calculator) and *nghanolwr* (mediator) in this list suggests that mutations undermine the word2vec method's ability to act as a means of clustering wordforms according to their semantic meanings. When trying to identify wordforms that have similar meanings to mutated wordforms in particular we see interference from the morpho-syntactic patterns found in the language's mutations being highlighted instead. This suggests that there is a need to develop methods of training vectors that ignore, or at least reduce, the influence of mutation on the model.

One possible solution would be to de-mutate the training corpus before using it to train a model, mapping the mutated wordforms to the embedding of the non-mutation form. That is, each *ddisgybl* and *nisgybl* could be converted into *disgybl* in the training data so that a universal vector could be ascertained for them, and then that universal vector could be set as an embedding for *disgybl*, *ddisgybl* and *nisgybl*.

We hope to experiment with this in the future using the TagTeg tagger (which enables lemmatization on the basis of a word's part of speech - see Prys and Watkins in this volume [22]), along with the Welsh Language Text Manipulation Tools that have also been released by the Unit.[5] Manipulation of other aspects, such as changing plural forms into the singular, may also be useful. However, the level of data manipulation required may be dependent on the exact NLP task being undertaken. Mutation information may be useful for a dependency parser, for example. As

[5]https: //github.com/techiaith/offer-trin-iaith

a result, this approach will have to be tested and evaluated. It may be that the best solution is to try to increase the size of the corpus in the hope that results for *nisgybl* improve. However, there may not be enough suitable Welsh language data available to enable such improvement. Mathematical operations to mitigate for mutation mahy also be possible.

As with the above results for the unmutated forms, the results for *coleg* (college) and *ysgol* (school) seem relevant and appropriate. In the case of *coleg* there are many different educational establishments listed (among them feithrinfa (nursery), ysgol (school), prifysgol (university), llyfrgell (library) and clwb (club), as well as the educational sessions held there such as cyrsiau (courses) and darlithoedd (lectures)).

Table 7: Wordforms similar to *coleg*

Wordforms similar to **coleg**	Similarity (L2 Norm)
brifysgol	0.7886645197868347
goleg	0.6492246985435486
prifysgol	0.6213444471359253
Brifysgol	0.6134623289108276
Coleg	0.6103212833404541
cwrs	0.5707718133926392
campws	0.5558643341064453
colegau	0.5230373740196228
feithrinfa	0.5063283443450928
Goleg	0.5032021999359131
ysgol	0.4937019646167755
cyrsiau	0.49133726954460144
darlithoedd	0.4851536750793457
myfyriwr	0.48434898257255554
llyfrgell	0.4834088385105133
darlithwyr	0.4713546931743622
myfyrwyr	0.4686380624771118
prifysgolion	0.4667356014251709
uwchraddedig	0.46337422728538513
clwb	0.4560936689376831

Table 8: Wordforms similar to *ysgol*

Wordforms similar to **ysgol**	Similarity (L2 Norm)
Ysgol	0.6604846119880676
ysgolion	0.6279251575469971
ysol	0.5601516962051392
feithrinfa	0.5393844842910767
ysgol-	0.5298383235931396
adran	0.52287358045578
eglwys	0.5205191373825073
athrawes	0.5143859386444092
hysgol	0.504888653755188
coleg	0.4937019646167755
athrawon	0.4809763431549072
athro	0.4706666171550751
ystafell	0.4681328237056732
iard	0.4598560929298401
addysg	0.4504145085811615
ardal	0.4488360285758972
hosbis	0.4486157298088074
amgueddfa	0.44594961404800415
aelwyd	0.4384314715862274
ysbyty	0.4344925284385681

In the case of *ysgol*, the results also seem appropriately relevant to the concept of an educational institution or social hub. The example of *ysol* is particularly interesting as it is clear from the context that the vector represents a misspelling of *ysgol*. This highlights one of word2vec's problems, which is that it cannot distinguish between when *ysol* is a misspelling of *ysgol* and when *ysol* represents an adjective which is related to *ysu* (consuming). That meaning seems to have been drowned out by the relationship between the misspelling of *ysgol* and similar wordforms from the field of education.

6.2 The Specific Problem of Homographs in Welsh Text

Misspellings such as *ysol* are not the only source of conflicts between homographs that share the same orthographic form but represent different concepts. Although not as frequent in Welsh as in English, there are many words in Welsh that can represent more than one meaning. These are referred to as *homonyms*, with *ysgol* being one example as it can represent both *a ladder* and *a school*. As we can see from table 8 above, the semantic meaning of 'a ladder'

has been completely drowned in the results by educational senses. This likely reflects the fact that there is more educational discourse in contemporary written Welsh than there is discussion of construction equipment, and underlines that some areas are less represented than other within the general Welsh-language discourse.

In addition to the ambiguity caused by radical forms such as *ysgol* that happen to represent multiple distinct concepts, further ambiguity is caused in Welsh due to the way in which the mutations can cause words which do not conflict in their radical forms to clash in their mutated forms [23]. Forms such as *law*, for example, may represent a mutated form of *glaw* (rain) or a mutated form of *llaw* (hand). In table 9, we see that the results are a mixture of word forms that have similar meanings to rain (gawodydd (showers), glaw (rain), wynt (wind), gymylau (clouds), lawiad (rainfall)) and hand-like ones (fraich (arm), gledr (palm), gern (cheek)).

Table 9: Wordforms similar to *law*

Wordforms similar to **law**	Similarity (L2 Norm)
gawodydd	0.499844491481781
glaw	0.4749141037464142
gern	0.40383607149124146
fraich	0.39562690258026123
wynt	0.39277803897857666
gledr	0.3897736072540283
forthwyl	0.38594579696655273
gymylau	0.3850449025630951
lawiad	0.38417601585388184
linoleum	0.38245636224746704

Table 10: Wordforms similar to is

Wordforms similar to **is**	Similarity (L2 Norm)
uwch	0.6420556902885437
Is	0.6010016202926636
Iselach	0.5591913461685181
Îs	0.5208320617675781
is-	0.508048415184021
was	0.4835599660873413
uchelach	0.47687333822250366
sub	0.4627269506454468
comes	0.4506301283836365
lower	0.44197165966033936

Within the context of the Welsh language, where there are often English phrases and titles, *law* may even represent some of the semantics of the English word *law*. Whilst that meaning may not be too apparent in these results, the influence the semantic meanings of Englishs wordforms can clearly be seen with other forms such as *is* (lower). As can be seen in table 10, although Welsh forms do come higher in the results, both *was* and *sub* (which appear sixth and eighth in the table) represent English forms, with *was* being similar in meaning to the English verb *is*. Although we have attempted to exclude English texts from the training corpus, it is inevitable that some English will be present as English often occurs within Welsh sentences in the form of titles and quotes.

6.3 An Evaluation of 'Brenin'

One of the canonical examples [24] used for evaluating the usefulness of embeddings as a means of converting lexical information into quasi-semantic numerical information is the one where is the ability to subtract the value of the vector for *man* from the vector value of the wordform *king* and return *queen* as the most similar wordform.

We were able to achieve similar results with our Welsh embeddings by subtracting the values for *man* from the values for *king* and adding the values for *woman*. The line of code used was:

```
show (model.most_similar (positive = ['king', 'woman'], negative = ["man"], topn = 10))
```

There is an example of this in the example.py file available in our GitHub repository. In table 11, we see that the three forms most similar to *brenin* (king) are *Brenin*, *tywysog* (prince) and *frenin* (mutated form of king). But *brenhines* (queen) does not appear in the results at all. In table 12, however, *brenhines* is the most similar word, and the mutated form, *frenhines*, is second.

Table 11: Wordforms similar to *brenin*

Wordforms similar to **brenin**	Similarity (L2 Norm)
Brenin	0.7919192910194397
tywysog	0.7350390553474426
frenin	0.6617879867553711
frenhines	0.6556856632232666
Pharo	0.6539126634597778
brenhinoedd	0.6522496938705444
Herod	0.6422858834266663
pab	0.6362754702568054
Frenin	0.6353232264518738
marchog	0.6326084136962891

Table 12: Wordforms similar to *brenin* having added *dynes' values* and subtracted *dyn's*

Wordforms similar to **brenin – dyn + dynes**	Similarity (L2 Norm)
brenhines	0.5559254884719849
frenhines	0.5395671129226685
Pharo	0.5325185656547546
Brenin	0.5258678197860718
brenhinoedd	0.5248633623123169
Dareius	0.4983474612236023
tywysog	0.4887065291404724
Rehoboam	0.4866292178630829
Dál	0.4801225960254669
Antiochus	0.477974534034729

We believe that the many examples of the names of kings and other leaders are due to the inclusion of the three different versions of the Bible in the training data, in addition to the data from Wikipedia which has a tendency to list the names of historical kings within its articles. Interestingly, by subtracting the vector of *Israel* from the vector of *brenin*, we were able to reduce the influence of Biblical names. This gave *tywysog*, *frenhines*, *brenhines*, *marchog* (knight), *barwn* (baron), *dywysoges* (mutated form of princess), *goron* (mutated form of crown), *porthor* (porter) and *telynor* (harpist) as the top ten results (including both capitalized and non-capitalized forms). This ability to manipulate results mathematically demonstrates the flexibility of word embeddings where wordforms can be represented as numerical values.

6.4 Evaluation of 'Rheoliadau' and 'Firws'

Thus far, we have taken a critical look at some of the problems that arise with the word2vec approach within the context of the Welsh language. Despite some minor issues, it is clear that the model remains a valuable resource. In the following sections we present results which were not designed to identify issues and which are therefore more representative of the performance of the model, and as a result better indicate the potential of this approach for Welsh language technologies. The following words were chosen in a more random manner to try to give a taste of its performance in a variety of different fields.

To demonstrate the model's performance with terms used in the field of governance, the term *rheoliadau* (regulations) was selected (see Table 13). We can see that the wordforms returned as being similar in meaning are all clearly relevant, and relate to principles to be followed within the context of governing society.

Health is another area where the Welsh language has been expanding increasingly in recent years. With the recent pandemic at the forefront of our minds, we chose to search the model for words most similar to *firws* (virus). The results obtained were all relevant, highlighting the fact that there are two recognized spellings for *virus* in Welsh, one that maps directly from the Latin form (i.e. *firws*), and another (i.e. *feirws*) that emulates *virus* as pronounced in English.

Table 13: Wordforms similar to *rheoliadau*

Wordforms similar to **rheoliadau**	Similarity (L2 Norm)
canllawiau	0.7212351560592651
deddfwriaeth	0.7080656290054321
darpariaethau	0.6954891681671143
deddfau	0.6948240995407104
deddfwriaethau	0.6894554495811462
cyfreithiau	0.6881170272827148
rheolau	0.6874794960021973
rheolaethau	0.6767716407775879
polisïau	0.6742568016052246
reoliadau	0.6652120351791382

Table 14: Wordforms similar to *firws*

Wordforms similar to **firws**	Similarity (L2 Norm)
feirws	0.7919204235076904
haint	0.741651177406311
firysau	0.6742879748344421
malaria	0.6648149490356445
bacteria	0.6618992686271667
clefyd	0.6611193418502808
bacteriwm	0.656657874584198
pathogen	0.6478495001792908
clwyf	0.6430660486221313
heintiau	0.6409652233123779

In an attempt to further challenge the model, it was tested with more unusual and complex wordforms.

6.5 Evaluation of 'Ebran' and 'Cofalent'

Ebran is an old name for horse feed. It was chosen to test the model on older agricultural vocabulary - vocabulary which would not be expected to be widely represented on the web. Although *ebran* is no longer a common word, the results, such as *cheirch* (oats), *wenith* (wheat), *maip* (turnips) and *rwdins* (swede) as seen in table 15 all apply in the context of animal feeds, with *borant* (they are grazing) and *fwytasant* (they ate) conveying the associated action. *Gascl* (stack) and *tewion* (fat ones) are relevant descriptive words too. To test a word from the other end of the spectrum, *cofalent* (covalent) was chosen. This term describes a type of chemical bond that holds atoms together in a molecule. The forms are all relevant, except for *onig*. This represents an encoding problem with the form *ïonig* (ionic). One could perhaps have expected to see *atomau* (atoms) higher up the list (it was 22nd in the complete list), but its position is probably due to the fact that *atoms* is not a qualifier, and in general there is a tendency for qualifiers correspond most closely to *cofalent* because *cofalent*, as an adjective, is itself a *qualifier*.

Table 15: Wordforms similar to *ebran*

Wordforms similar to **ebran**	Similarity (L2 Norm)
gascl	0.498212069272995
cheirch	0.49560439586639404
tewion	0.49511802196502686
ymenyn	0.49272191524505615
maip	0.49177083373069763
fwyttasant	0.47371983528137207
rwdins	0.47243034839630127
wenith	0.4720260500907898
borant	0.47085273265838623
fwsg	0.4696589708328247

Table 16: Wordforms similar to *cofalent*

Wordforms similar to **cofalent**	Similarity (L2 Norm)
ïonig	0.844214677810669
Hydrogen	0.7465869188308716
onig	0.7396742701530457
peptid	0.7186888456344604
gofalent	0.7003843784332275
alcen	0.695235013961792
moleciwl	0.6931548118591309
molecylau	0.6825446486473083
niwclews	0.6779775619506836
hydrogen	0.673660933971405

6.6 Evaluation of 'Maswr' and 'Cwrw'

To vary the context a little, we chose next to examine the results for two terms from more popular fields.

Table 17: Wordforms similar to *maswr*

Wordforms similar to **maswr**	Similarity (L2 Norm)
mewnwr	0.8384682536125183
blaenasgellwr	0.8043328523635864
cefnwr	0.7888342142105103
bachwr	0.7347344756126404
canolwr	0.7106105089187622
asgellwr	0.6771109104156494
wythwr	0.6747544407844543
blaenwyr	0.6649476289749146
prop	0.6608309745788574
blaenwr	0.6525115966796875

Table 18: Wordforms similar to *cwrw*

Wordforms similar to **cwrw**	Similarity (L2 Norm)
gwin	0.7391205430030823
seidr	0.7334795594215393
gwirodydd	0.718646764755249
coctels	0.687247633934021
jin	0.6688615083694458
gwrw	0.6675115823745728
chwrw	0.6663733124732971
lager	0.6653351783752441
lemonêd	0.6433229446411133
melysion	0.6420143842697144

The first, *maswr* (outside-half), is a word from the world of rugby, where it represents a specific position held by a player. The results here were very relevant, with each similar wordform also representing a player's position on the field.

The next word was *cwrw* (beer), and again the results were relevant and largely belonged to the same category, namely that of alcoholic beverages. The next items in the list were *lemonêd* (lemonade) and *melysion* (sweets) showing the results veer away slightly in terms of relevance but remained relevant to refreshments.

6.7 Evaluation of 'Gwarged' and 'Gwaddol'

We also chose to test the results for *gwarged* (surplus) and *gwaddol* (endowment), two words which are somewhat less common without being completely unfamiliar.

Table 19: Wordforms similar to *awaraed*

Wordforms similar to **gwarged**	Similarity (L2 Norm)
warged	0.6837567090988159
enillion	0.6060212850570679
refeniw	0.5929793119430542
trosiant	0.5913615822792053
gorwariant	0.5776404142379761
gorbenion	0.5762640237808228
gros	0.5747412443161011
net	0.5730563402175903
elw	0.5707241296768188
drosiant	0.5644094347953796

Table 20: Wordforms similar to *gwaddol*

Wordforms similar to **gwaddol**	Similarity (L2 Norm)
cyfraniad	0.41669243574142456
etifeddiaeth	0.4150990843772888
cymynrodd	0.41348540782928467
gweledigaethau	0.37813234329223633
waddol	0.3666684329509735
gwreiddiau	0.3642991781234741
creithiau	0.36246055364608765
traddodiadau	0.3608437478542328
Moderniaeth	0.35934978723526
traddodiad	0.35840266942977905

The results for *gwarged* all relate to the field of finance, but one might have expected to see *gweddill* (remainder) appear higher in the results. The results for *gwaddol* are interesting because they reflect that the word is used metaphorically for abstract concepts which can also be inherited, such as 'tradition' and 'roots', in addition to more material types of inheritance. It is also interesting to note that while the results for *gwaddol* are very appropriate, the similarity scores are comparatively low.

6.8 Evaluation of 'Isio' and 'Eisiau'

Embeddings are very useful in the context of texts that contain 'non-standard' forms, especially those which are more characteristic of spoken or dialectal forms. Below we see results for words similar to *isio* (a non-standard form of 'want') and words similar to *eisiau* (the standard form of 'want'). Note that the results for the standard form *eisiau* contain similar standard forms, while the results for the more colloquial form *isio*, contain mostly colloquial forms.

Table 21: Wordforms similar to *isio*

Wordforms similar to **isio**	Similarity (L2 Norm)
eisio	0.845168948173523
ishe	0.773231029510498
isho	0.7355859875679016
angan	0.6781127452850342
eisiau	0.6601892709732056
licio	0.6232104897499084
eisau	0.6142325401306152
moyn	0.6039586067199707
mhoen	0.5832076072692871
goro	0.5790119171142578

Table 22: Wordforms similar to *eisiau*

Wordforms similar to **eisiau**	Similarity (L2 Norm)
eisio	0.7607351541519165
isio	0.6601892709732056
angen	0.6397089958190918
dymuno	0.5431622266769409
wedi	0.5423191785812378
ishe	0.5412291884422302
gallu	0.5405140519142151
hoffi	0.5342394113540649
bwriadu	0.5273668766021729
agen	0.5158362984657288

Not only do these vectors allow systems to potentially cope with forms more similar to those used in speech, they are also a useful resource for researching dialect and for normalizing spoken forms by converting them to their more standard equivalents.

6.9 Evaluation of 'Anime', 'Cartwnydd' and 'Cartŵn'

Finding really poor and irrelevant results was challenging. We were a bit surprised to see such good results for *anime*, for example, as anime does not seem to be a topic that is often discussed through the medium of Welsh. Conversely, the results for *cartwnydd* (cartoonist) were a little disappointing, with the well-known cartoonist Huw Aaron's surname and the equivalent English term *cartoonist* appearing higher in the table than words which are more similar in their meaning.

Table 23: Wordforms similar to *anime*

Wordforms similar to **anime**	Similarity (L2 Norm)
manga	0.6116302013397217
animeiddiedig	0.5045759081840515
rhaghysbyseb	0.4528508484363556
animeiddio	0.4499656558036804
Avatar	0.4427505433559418
Lolita	0.4420943260192871
Anime	0.4337618350982666
archarwyr	0.43363142013549805
animeiddiadau	0.43353867530822754
Marvel	0.43206268548965454

Table 24: Wordforms similar to *cartwnydd*

Wordforms similar to **cartwnydd**	Similarity (L2 Norm)
Aaron	0.4788210093975067
cartoonist	0.4635165333747864
Cartŵn	0.4398535192012787
gauntlet	0.43284767866134644
Onllwyn	0.41497212648391724
Bebbteirawr	0.40730375051498413
ffotograffydd	0.403152734041214
Chiswell	0.3967725932598114
Brassington	0.3945808708667755
gartwnau	0.3921811282634735

However, the results for *cartŵn* (cartoon) were much better, demonstrating that it is difficult to predict where weaknesses may appear in the model.

Table 25: Wordforms similar to *cartŵn*

Wordforms similar to **cartŵn**	Similarity (L2 Norm)
cartwn	0.81761634349823
comig	0.7451732158660889
gartŵn	0.6066957712173462
cartwnau	0.6000196933746338
montage	0.5497390031814575
portread	0.5466684699058533
paentiad	0.5457241535186768
darluniadau	0.5372176170349121
collage	0.5316237211227417

Table 26: Wordforms similar to *arafu*

Wordforms similar to **arafu**	Similarity (L2 Norm)
cyflymu	0.7991060018539429
gyflymu	0.5817015171051025
stopio	0.5545011758804321
arafi	0.5365293025970459
ymledu	0.5314207077026367
sefydlogi	0.5304709076881409
erydu	0.5254966616630554
gwaethygu	0.5129178762435913
lleihau	0.5124251246452332

6.10 Evaluation of 'Arafu'

A widely recognized issue with word vectors is that antonyms such as *happy* and *sad* are often identified as having similar meanings, even though they represent two different semantic poles [25]. This is because antonyms often occur within very similar sentences. For example, in sentences such as 'I feel [word]' or 'she was [word] today', [word] could be swapped for either *sad* or *happy* without the sentence appearing odd or unusual. In table 26, above, we see an example of this problem within wordforms that, according to the model, are supposed to be similar to the wordform *arafu* (to slow). The first result (which according to the model is by far the most similar) is *cyflymu* (to accelerate), which is the complete opposite in meaning to *arafu*. This is a clear demonstration that the word2vec method is not a true semantic method, but rather one that relies on identifying words that are used in a similar manner. Perhaps in our case, the fact that the corpus contains many educational sentences that handle *arafu* and *cyflymu* in similar, formulaic ways has contributed to highlighting this problem. Nevertheless, this is a known issue with embeddings in general, and one that users need to be aware of.

6.11 OOV Words

The ability of the word embeddings method to deal with out of vocabulary (OOV) words is another major problem. We experimented fairly haphazardly with different words to try and 'trip up' the model by suggesting unusual words for it to find similar corresponding words. We found it difficult to think of valid Welsh words that were not already included in the model, and not for lack of imagination. Even words such as *interniaeth* (internship), *hunanynysu* (self-isolation) and *hunlun* (selfie), words that were not in our dictionaries' lexicon at the time, were to be found within the model. We had more 'luck' with terms newly coined as part of the Unit's terminology

standardization work, *adgyfogi* (regurgitation) being a good example of such a term. The term had only been recently coined *adgyfogi* as a way of distinguishing between *cyfogi* (vomiting) and regurgitation in medical texts.

Although it would seem that the model is therefore generally comprehensive, it is inevitable that a word2vec model will not include every possible wordform. Not only do languages change and add new words, new product names and different misspellings appear in real-life texts that will not be represented as embeddings in a model. The issue here is that each word in a sentence must usually be assigned some form of value for the systems that make use of vectors (for example within neural machine translation systems) to function. If it is not possible to derive an appropriate value from the model for the embedding, it is often necessary to use some other strategy to obtain a dummy or substitute value for the wordform. While it is better to have a value that is as appropriate as possible, using any value is usually better than not having any value whatsoever as the system requires a value to work. There are a number of different strategies for creating an embedding for a wordform where none is available. Dictionary lists can be used to try to identify synonyms so that the synonym's value can be used. Another method is to try to identify similarly spelled wordforms and use their embeddings. However, there is a limit to the ability of those methods to find semi-appropriate embeddings. The latest solution to this problem is to use methods that create embeddings for subword tokens (that is, parts of wordforms) so that a vector for OOV words can be created out of a combination of two or more subword embeddings. This may not always produce a highly appropriate embedding (the meaning of words cannot always be deduced by looking at their elements), but it has been shown to work well in many cases, and importantly ensures that a value exists for each lexical unit within the text. We hope to evaluate this method in the near future by training a FastText model on the training corpus (doing so in Gensim is a relatively simple matter after having already trained a word2vec model), so that the results can be compared.

6.12 Benefits to the Pipeline and to Terminology Work etc.

In evaluating the above results we have focused on analysis from the human linguistic perspective. In that context, it is clear that the results would be useful for human lexical and terminological research. In an attempt to predict the value of the model with respect to more automated processes, the model was used to strengthen the training of an early version of the Welsh-language dependency parser we have trained using spaCy. The results suggest that the vectors provide a ~10% improvement to that parser, but this will need to be confirmed once our work on dependency parsing is more advanced.

7 CONCLUSIONS

In summary, we have presented initial findings that show that a large corpus of Welsh texts can be collected and used to train a word2vec model that provides truly useful results. Our qualitative analysis suggests that it does so for words that belong to a variety of subject areas and different registers and styles, even when the forms are relatively uncommon. There instances where the model does not have a vector for a valid Welsh word are few and far between, and irrelevant results for the most similar words are rare.

However, we have also identified some areas for improvement. As we have seen, distinguishing between the different meanings of polysemous wordforms is not a strength of word vector models. An example of the antonym problem which effects word2vec methods also raised its head, with *arafu* and *cyflymu* being suggested as similar despite being polar opposites. This suggests that there is scope to increase the size of the corpus, and to ensure that it contains richer sentences that place these antonyms in different contexts. That might mean focusing on adding more literary sentences rather than collecting more web texts and educational materials. We also believe that there

is scope for adding texts from less represented subject areas, as well as increasing the number of texts written in rarer styles and registers, not to mention increasing the inclusion of textual representation of spoken Welsh. This in turn means gathering more data from Welsh-language organizations, including public bodies and private presses, in order to improve the models available to everyone.

Despite the generally high standard of the results, we have recognized that Welsh-language mutations pose an additional problem for Welsh word vector models as they disperse semantic information between a number of additional forms where there is in truth no actual semantic difference. We believe that this requires de-mutating or de-inflecting the training corpus using software that can do so appropriately, converting, for instance, *ceir* as a verb to *cael*, and *ceir* as a plural noun to car. Our TagTeg tagger is promising in this regard. It will be necessary to experiment to find out which types of de-inflection are suitable, but our hypothesis is that it will be worthwhile for more semantically orientated tasks, but will perhaps be less appropriate where there are more syntactic aspects to the required use case.

Another task to be undertaken is to experiment with adjusting the training parameters of the wor2vec model, and to attempt to train a Fasttext model to evaluate whether or not results are significantly improved by using more sophisticated methods of producing embeddings (ones that operate on the basis of parts of words or subwords rather than whole wordforms). However, our general feeling is that increasing the quantity and variety of training data will have the greatest improvement in the standard of models, and that doing so should be our main priority.

We hope that this work will be useful not only to facilitate human language research, but also to improve the standard of linguistic components and processes in Welsh where they could benefit from the knowledge contained within word vector models.

ACKNOWLEDGMENTS

We are grateful to the Welsh Government for funding this work as part of the Text, Speech and Translation Technologies for the Welsh Language 2020 – 2021 project.

REFERENCES

[1] Tomas Mikolov, Kai Chen, Greg Corrad and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In 1st International Conference on Learning Representations (ICLR 2013). Arizona, USA, 1-12.

[2] Delyth Prys, Gruffudd Prys and Dewi Jones. 2016. Cysill Online: A Corpus of Written Contemporary Welsh Compiled from an On-line Spelling and Grammar Checker. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Portorož, Slovenia, 3261-3264.

[3] Geraint I. Palmer. Projects. Accessed April 12, 2021 from http://www.geraintianpalmer.org.uk/projects/

[4] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013).

[5] Ignatius Ezeani, Scott Piao, Steven Neale, Paul Rayson and Dawn Knight. 2019. Leveraging Pre-Trained Embeddings for Welsh Taggers. In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019): Held at the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy. Association for Computational Linguistics, Florence, Italy, 270-280.

[6] Delyth Prys, Mared Roberts and Gruffudd Prys. 2016. Reprinting scholarly works as e-books for less-resourced languages. In Proceedings of the Tenth International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Portorož, Slovenia, 74-79.

[7] Dewi Jones, Gruffudd Prys and Delyth Prys. 2016. Vocab: A dictionary plugin for websites. In Second Celtic Language Technology Workshop. Paris, France.

[8] Tegau Andrews and Gruffudd Prys. 2016. Terminology Standardization in Education and the Construction of Resources: The Welsh Experience. Education Sciences 6, 2 (January 2016), 1-15.

[9] Dewi Jones, Patrick Robertson, and A. Taborda. 2015. Corpus of Welsh Language Tweets. Accessed April 12, 2021 from http://techiaith.cymru/data/corpora/twitter/?lang=en

[10] Thomas Mayer and Michael Cysouw. 2014. Creating a Massively Parallel Bible Corpus. In Proceedings of the Ninth International Conference

on Language Resources and Evaluation (LREC 2014). European Language Resources Association (ELRA), Reykjavik, Iceland, 3158-3163.

[11] The Bible Society. 2015. Y Beibl Cymraeg. Accessed April 12, 2021 from https://www.biblesociety.org.uk/cymru/y-beibl-cymraeg/?cymraeg

[12] Ellis, C. O'Dochartaigh, William Hicks, M. Morgan and N. Laporte. 2001. Cronfa Electroneg o Gymraeg (CEG): A 1 million word lexical database and frequency count for Welsh. [Online].

[13] Kevin Donnelly and Margaret Deuchar. 2011. The Bangor Autoglosser: A Multilingual Tagger for Conversational Text. In Proceedings of the Fourth International Conference on Internet Technologies and Applications (ITA 11). Glyndwr University, Wrexham, North Wales, 17-25.

[14] Pedro Javier Ortiz Suárez, Laurent Romary and Benoît Sagot. 2020. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, 1703-1714.

[15] Dawn Knight, Steve Morris, Tess Fitzpatrick, Paul Rayson, Irena Spasić, Enlli-Mon Thomas, Alex Lovell, Jonathan Morris, Jeremy Evas, Mark Stonelake, Laura Arman, Josh Davies, Ignatius Ezeani, Steve Neale, Jennifer Needs, Scott Piao, Mair Rees, Gareth Watkins, Lowri Williams, Vignesh Muralidaran, Bethan Tovey-Walsh, Laurence Anthony, Thomas M. Cobb, Margaret Deuchar, Kevin Donnelly, Michael McCarthy and Kevin Scannell. 2020. CorCenCC: Corpws Cenedlaethol Cymraeg Cyfoes – the National Corpus of Contemporary Welsh Dataset. http://doi.org/10.17035/d.2020.0119878310.

[16] Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics 5, (2017), 135-146.

[17] Kevin Patel and Pushpak Bhattacharyya. 2016. Towards Lower Bounds on Number of Dimensions for Word Embeddings. In Proceedings of the 8th International Joint Conference on Natural Language Processing. Asian Federation of Natural Language Processing, Taipei, Taiwan, 31–36.

[18] Bakarov, Amir. 2018. A survey of word embeddings evaluation methods. arXiv preprint arXiv:1801.09536 (2018).

[19] Anonymous. 2020. Assessment of Deliverables (Welsh Language Technology). Evaluation Report.

[20] Martin J. Ball and Nicole Müller. 2002. Mutation in Welsh. Routledge.

[21] John R. Firth. 1957. A synopsis of linguistic theory 1930–1955. Studies in Linguistic Analysis: 1–32. Reprinted in F.R. Palmer, ed. 1968. Selected Papers of J.R. Firth 1952–1959. Longman, Llundain.

[22] Gruffudd Prys and Gareth Watkins. 2021. Developing a Part of Speech Tagger and a Corpus of Training Sentences for the Welsh Language. Language and Technology in Wales Volume 1. Bangor University, Bangor, Wales.

[23] Martin J. Ball. 1990. Homonymic clash and initial lenition in Welsh: 1, (000), 000,000 problems. Word 41, 3 (1990), 329-335.

[24] Levy, Omer, and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In Proceedings of the eighteenth conference on computational natural language learning, 2014. Association for Computational Linguistics, Ann Arbor, Michigan, 171-180.

[25] Bruna Thalenberg. 2016. Distinguishing antonyms from synonyms in vector space models of semantics.

Implementing NMT at a Welsh translation company

Challenges and findings of a project to embed neural translation technology at a Welsh translation company

MYFYR PRYS

Cymen Cyf , Caernarfon, Wales

This chapter describes the results of a one-year partnership between Bangor University, the Welsh Government, and the Welsh translation company Cymen Cyf to develop and embed neural machine translation technology in Cymen's workflows and technological infrastructure. We used the open-source software package Marian NMT to train neural translation engines that could be embedded in the company's translation tools. The relative quality of these engines was measured using the BLEU metric, which allowed us to observe a 5-point increase in quality of the engines after tuning the model hyperparameters. The optimized version of our engine performs 9 BLEU better than Google Translate on Cymen's own internal data, and 5 BLEU better on a 'neutral' external dataset. As well as developing the engines themselves, the project also involved developing an app to leverage the technology in a computer-assisted translation tool.

Key words and phrases: neural machine translation, the Welsh economy, translation memories

1 INTRODUCTION

The translation sector in Wales has recently been identified by the Welsh Government as a crucial area for development [1]. The past few decades have seen a gradual growth in demand for English to Welsh translation, due to successive waves of legislation intended to safeguard the Welsh language [2]. In its most recent strategy paper Cymraeg 2050: A Million Welsh Speakers [1], the Welsh Government describes the need for "a modern and responsive translation profession which makes full benefit of the latest technology, and language resources (dictionaries, terminologies, and corpora) [...]". The 'latest technology' here can be understood primarily as an allusion to translation memory and machine translation technology, two innovations that have the potential to substantially extend the productivity of a single human translator. Research by Screen [3] has underlined the potential of adopting new technology to increase productivity in the Welsh translation industry. His work focusing on translation memories and machine translation estimates that using machine translation technology could raise the productivity of a group of eight translators from roughly 2.5 million words to around 7 million words a year.

In a previous paper, Prys and Jones [4] describe the implementation of machine translation technology at a North Wales translation company, Cymen Cyf. That project was part of a Knowledge Transfer Partnership (KTP) which is a tool used by the UK Government to foster links between the research sector and private industry. The rationale for the partnership was to increase productivity at the company by embedding bespoke, domain-specific translation engines in the company's workflows. A further goal was the transfer of academic knowledge to the company and permanent upskilling of staff, leading to a positive knock-on effect for the sector as a whole. That project was considered a success and was awarded the highest grade by the funding body, Innovate UK. This chapter reports on a follow-up project, a SMART Partnership between Bangor University, Cymen and the Welsh Government. Whereas the KTP implemented statistical machine translation technology (SMT), the new SMART Partnership has focused on upgrading to the new neural machine translation (NMT) paradigm. The next section will briefly introduce machine translation and translation memory technology, before moving on to describe how the project was implemented.

2 TRANSLATION TECHNOLOGY

Translation technology spans a broad variety of approaches. In this chapter, I will focus on two technologies most relevant to the SMART partnership: translation memories and machine translation.

2.1 Translation Memories

Translation memories are essentially databases allowing previously translated material to be re-used. They are generally used in the context of translation software. Figure 1 below shows a translator using a translation memory to post-edit a document in Trados Studio.

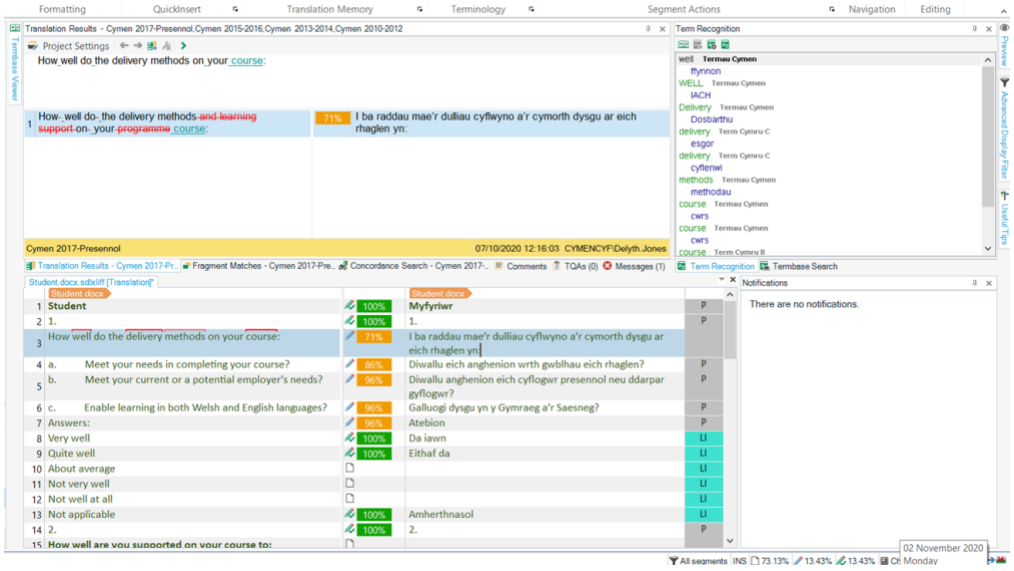

Figure 1 – A translation memory being used to post edit a document in Trados Studio

A post-editing workflow using translation memories usually works as follows. As a translator works on a document, the translation software checks each segment against the translation memory. If the translation memory contains a similar segment, it will flag this for the translator along with visual cues conveying the extent of the editing work necessary. In figure 1 for instance, segments have been matched to varying degrees. Some are complete matches and are represented in dark green. The translator may decide to leave these segments as they are. Other segments are partial matches, with the percentage of the match appearing in orange. The translator will usually make edits to these segments using the software's set of visual cues as a guide. In the main editing window in figure 1, Trados is suggesting that the user delete the red text and add the aqua coloured text.

There is large body of research demonstrating the advantage of translation memories to translators, and translation companies [3]. But translation memories have important implications beyond this too, as they are an ideal storage medium for parallel texts essential for statistical and neural machine translation.

2.2 Machine Translation

Machine translation is a technology that allows a user to translate a text automatically, even if that text has never been seen by the system previously. The way that this is done varies considerably, but the three main paradigms have been rule-based machine translation, statistical machine translation and neural machine translation. Rule-based machine translation was the initial paradigm of machine translation. The technique involves programming explicit grammatical rules which determine how the engine will translate from one language to another. The main drawback for these types of engines is their lack of flexibility, and the requirement of a substantial investment in time for creating systems [5].

Statistical machine translation (SMT) is part of the second wave of translation technology. Rather than using rules programmed by hand, SMT models learn patterns from being trained on parallel corpus data. In the most successful approach, phrase-based statistical machine translation, phrasemes that co-occur statistically in the data are scored and extracted into massive phrase tables, which can then be re-ordered to create a final translation. The SMT era was the first in which machine translation was considered good enough to be used on an industrial scale, with MT becoming part of many companies' workflows and translators being trained to post-edit [6]. Among the drawbacks for SMT models are their relative lack of fluency (compared with NMT), the complexity of the model architecture and the large memory footprint of phrase tables. SMT models were the state of the art in machine translation until the coming of NMT in the mid-2010s.

Neural machine translation is the most recent wave of machine translation. Rather than using phrase tables, NMT models use neural networks. The model comes in two parts – an encoder and a decoder – which both contain layers. The segment to be translated is first processed word by word by the encoder, which converts each word to a vector. Vectors are sequences of numbers which can represent extraordinarily rich information about each word in several dimensions, allowing the model to make more effective predictions. One of the primary advantages of NMT is that it can produce translations that are much more appealing to human evaluators [7], [8]. Part of the reason for this is that neural translation engines are significantly less likely to produce grammatical errors, with Bentivogli et al. [9] showing that they produce 19% fewer word errors, 17% fewer morphological errors and 50% fewer word order errors. However, NMT models seem to require more training data than SMT models to perform effectively [10]. They may also be more adversely affected by poor quality training data [11].

In order to make choices in terms of which translation engines should be used, a reliable method of measuring MT quality is necessary. Until the beginning of the twenty first century the most common technique was to use human judges, although this was often a slow and costly process [11]. At present the most used metric is BLEU [12]. BLEU works by comparing a translation engine's output – the hypothesis - with human translations of the same data. A score of zero represents a hypothesis that bears no resemblance to the human reference translation whatsoever, while a score of 100 would indicate a hypothesis matching the human reference exactly. According to Wołk and Mazarek [13], a BLEU score between 15-30 may be considered a decent translation, while scores over 50 are considered excellent. BLEU scores are not considered comparable between different languages, and there is some evidence that the metric tends to underrate NMT systems in comparison with SMT [14].

2.3 Machine Translation in Minority Contexts

Previous work on machine translation technology in the Welsh context has been relatively limited. Tyers and Donnelly [15] developed a rule-based cy > en translation engine using the Apertium framework.[1] The researchers report scores of 15.68 and 32.21 BLEU for their cy > en translation engine when translating test sets from the Proceedings of the Welsh Assembly and the Welsh Wikipedia, respectively. Jones and Eisle [16] have described the development of an SMT engine trained on a parallel corpus extracted from the Proceedings of the Welsh Assembly. The engine was trained on 510,813 segments and achieved a score of 36.17 BLEU for the en > cy language direction and 40.22 BLEU for cy > en. More recently Prys and Jones [4] developed a series of domain-specific translation engines for the translation company Cymen, with bespoke engines trained on data belonging to specific clients. Results varied based on the size of the clients' training data, ranging from 48.53 for the client with the smallest data set to 59.06 for the largest. This study is described further in section 4 below.

Outside of the Welsh context, work carried out by researchers on the en <> ga language pair (English and Irish) offer an interesting comparison. Dowling et al. [17] describe work on *Tapadóir*, an en > ga SMT engine intended to facilitate translation for the Irish Government department responsible for the Irish language. The researchers achieved a BLEU score of 43.08 by combining domain specific translation memories with other publicly available resources. More recently the same team reported that an initial attempt to train an NMT engine underperformed *Tapadóir* by 6.40 BLEU points [10]. The researchers explained this result as being related to the relatively complex morphology of Irish coupled with scarce training data, which is a common problem in minoritized language contexts.

A follow up study by Defauw et al. [18] addressed these problems with a three-pronged data-gathering strategy. First the researchers scraped additional parallel data from bilingual websites using the Scrapy[2] tool and aligned these using Malign. Secondly, the researchers used a new publicly available parallel corpus available as part of the ParaCrawl [19] repository.[3] Finally, a synthetic corpora was created by back-translating monolingual corpus data. All these additional resources required cleaning using the Bicleaner tool to remove poorly aligned or otherwise problematic sentences. The researchers found that an en > ga model trained with a mixture of web-scraped and ParaCrawl data improved by 11 BLEU points on a domain specific test set and by 8.7 on a generic, open domain one. These findings indicate that although a large amount of data is necessary for training effective NMT models, web scraping and other data augmentation strategies may be a viable solution for low resource, minoritized language contexts.

3 CYMEN – AN INNOVATIVE WELSH TRANSLATION COMPANY

Cymen was established in 1987 as part of a wave of private translation companies appearing in response to a growth in demand for English to Welsh translation at the time [2]. Since that time, Cymen has grown to employ 24 members of staff along with over 20 external freelance translators. The company's offices are located in Caernarfon in North Wales, but its customers are located all over the world. A pattern that has come to define Cymen is a tendency to take a close interest in technological development. This has manifested itself in a series of joint research projects with Bangor University taking place between 2000 and 2020. The first project was carried out in 2000 with

[1]The engine can be accessed through a browser at https://www.apertium.org

[2]https://scrapy.org/

[3]https://paracrawl.eu/

the aim of establishing the use of translation memories at the company. The second project, beginning in 2017, was a two-year partnership aiming to establish the use of statistical machine translation (SMT) technology. The objective of the most recent project, starting in 2019, was to replace these SMT engines with NMT ones.

It should be emphasized that this kind of positive approach towards technology is not necessarily widespread in the translation industry. According to one recent survey, only 22% of UK translators reported that they were using translation engine technology, of which 70% claimed to be using Google Translate [20]. This raises an important issue – why should a company like Cymen bother with creating a relatively complex machine translation workflow if it is possible instead to use Google Translate or Microsoft Translator in Trados? There are several answers to this question. The first is that using these services invariably means sending information out of company premises into the hands of international third parties. This could have problematic legal repercussions for any company required to deal with sensitive or confidential documents. The second answer has to do with both the quality and flexibility of the machine translation service. Google and Microsoft each provide generic translation engines which are intended to handle the broadest variety possible texts for translation. By leveraging their own archives of previous translations however, translation companies can generate translation engines tailored for the text types they are most likely to translate.

4 CYMEN AND BANGOR UNIVERSITY'S PREVIOUS PARTNERSHIPS

Cymen's current translation memory workflows were first put in place during the initial KTP project in 2000. Since that time, the system has developed to be simple but effective. Each document that is to be translated for a client is associated with specific translation memories and glossaries relevant to that customer. The document is then converted into a project for translation in Trados, with all the relevant translation resources appearing automatically for the translators as soon as they begin working. This system is managed with a series of templates which set the configuration of projects generated for each client. Over time the company has built up a large store of translation memories that are specific to certain clients, as well as more general translation memories that are used for less regular clients.

A second KTP project was started in 2017. The aim of this second partnership was to take advantage of the company's large archive of previous translations to train SMT translation engines and to implement these in the company's workflows. This process is described in detail in Prys and Jones [4], but the following is a short summary.

The first choice in the project concerned the best way to convert the company's archive of previous translation into a form appropriate for machine translation. We decided to leverage the company's supply of translation memories for this purpose (see section 2.1). Translation memories are stored in the form of TMX files, which use XML markup to structure the data. A code pipeline was written in order to automate the process of extracting each parallel segment from Cymen's translation memories and build up a training corpus automatically. Further code was added in order to clean and prepare data for training. Among other things, this process involves removing XML markup and any target translations that were substantially longer or shorter than the source segment. The result is a parallel corpus that is clean and ready for the training process.

Training was facilitated by our ability to take advantage of previous work by Bangor University's Language Technology Unit, an SMT training pipeline using Moses SMT. This package is freely available at the Unit's Github

page.[4] Some additions made during the project were steps for tuning and automatic evaluation using BLEU. Following this, an app was developed to introduce the engines into the translation environment used by the company's translators. Trados's manufacturers have enabled the development of open-source plugins using code templates and pre-set projects available on their Github page.[5] One of these pre-set plugins[6] was adapted to create our custom plugin.

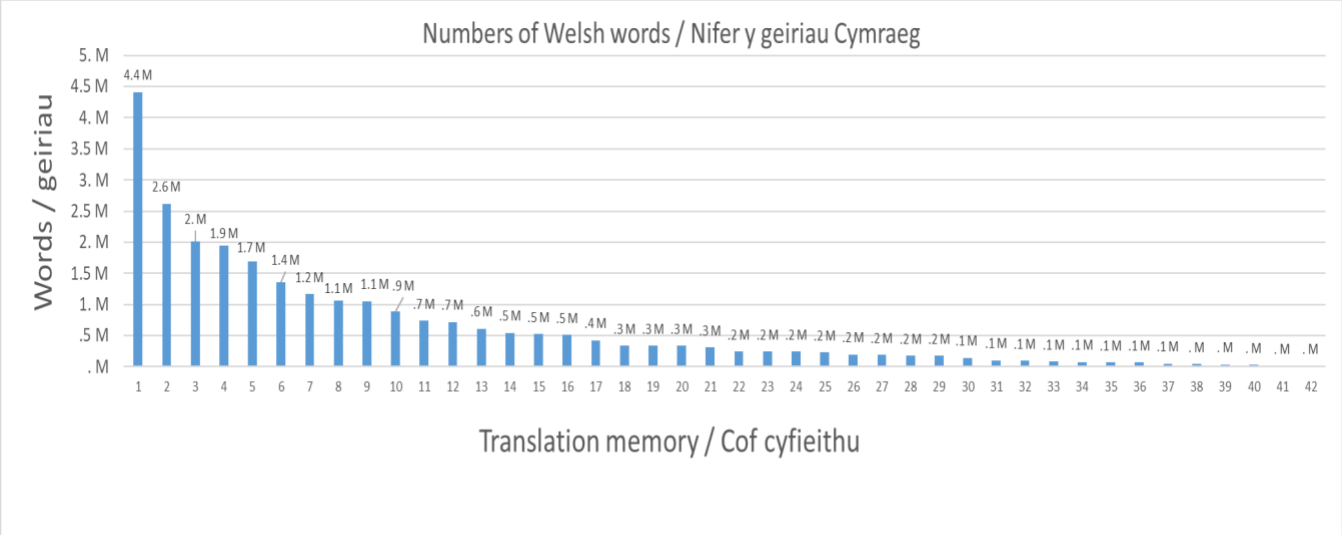

Figure 1 – The number of millions of words in Cymen's translation memories (2020)

The company's use of the engines began gradually, with management initially testing them before allowing their use more broadly. Any translation engines that did not seem to be effective were dropped immediately. In the end only two engines survived this process, those trained on translation memories 1 and 2 (see figure 1 above). The engine trained on TM 1 was found to be especially useful by the translators. The success of this particular translation engine was largely due to the fact that it satisfied two of machine translation's main criteria – highly relevant data along with a large training set [6]. This engine was set up to appear automatically for translators in all translation projects for this client, and feedback from translators was positive. The quality of the general domain translation engine (that is, an engine trained on all available data) was not considered useful in most cases and was largely unused.

5 SMART PARTNERSHIP AND NMT

Following the second KTP project, we were informed that a further partnership to upgrade the translation engines had been approved.

[4]https://github.com/PorthTechnolegaulaith/moses-smt

[5]https://github.com/sdl/Sdl-Community

[6]https://github.com/OpenNMT/Plugins

The first step of the project was to select a software package for training NMT models. After a short period of research, Marian NMT[7] was selected as it offers many innovative training features along with clear and accessible documentation. Following this, time was spent gathering additional training data. One intention that we had for this project was to attempt to develop a general domain engine capable of handling any client's data to a satisfactory level. To that end we decided to try and collect as much Welsh <> English data as possible from various sources.

Table 1: Training data for Cymen's general domain engine

Data	Number of words (Welsh)
Cymen's translation memories	89 million
Proceeds of the National Assembly + Legislation + Software	18 million
The Welsh Government's translation memories	2 million
The Welsh Government's glossaries	0.4 million
Total	109.4 million
Total after cleaning	86 million

As can be seen in table 1 above, additional data was collected from a variety of sources to complement the company's own data sets. The primary source was Cymen's internal archive of translation memories, which have grown to 89 million Welsh words since the previous project. Part of this growth came from old TMX files (as well as other file formats) that were discovered in Cymen's archives. Yet more was gathered in the course of the company's daily translation work, with around 1 million words being added to Cymen's translation memories in each month of 2020.

The second source was http://techiaith.cymru. This website provides three parallel corpora as part of the Language Technology Unit's Moses SMT implementation: a corpus extracted from the proceedings of the Welsh Assembly, a corpus of bilingual UK Government legislation, and a corpus of software translations. Finally, the Welsh Government have recently began to offer TMX files on their website for public use at https://llyw.cymru/bydtermcymru. This repository offers a large variety of TMXs relating to domains such as health, education and legislation. At this time (November 2020) the files contain around 2 million Welsh words in total, but the provision seems to be growing rapidly. The website also offers a number of glossaries standardised by the Welsh Government available in Excel file format that are easy to convert to parallel corpora. Additional open-source corpora for Welsh <> English exist at the Opus[8] repository, however we decided not to use these due to the extremely low quality of the data.

After combining these corpora and removing duplicate segments, the result is a relatively substantial corpus of 109 million Welsh words. Following cleaning (described in section 4) the corpus is reduced to around 86 million words. Considering the relatively small size of publicly available corpora for Welsh (see table 1), Cymen's situation appears to be particularly promising in terms of training NMT engines.

5.1 Configuring the NMT Model

A particularly important element in training NMT models is tuning the hyperparameters, elements of the model that can be configured by the researcher. These include regularization methods such as dropout [21] and

[7]https://marian-nmt.github.io/

[8]http://opus.nlpl.eu/

exponential smoothing, which help prevent the model from overfitting to the training data. Researchers can also select elements of the system architecture, with Marian allowing a choice between a basic RNN model, a seq2seq model and a transformer model [22] as well as the dimensions of the decoder and encoder. Other features like layer normalization [23] allow the training time of the model to be reduced.

Over a period of several experiments, we tested several different configurations, using BLEU scores as a guide to the efficacy of any changes. Some of the hyperparameters for the optimal model obtained are presented in table 2 below.

Table 2: Configuration of Welsh <> English general domain model

Feature	Value
Model architecture	Transformer
Source dropout	0.1
Target dropout	0.1
Encoder depth	4
Decoder depth	4
Layer normalization	-
Exponential smoothing	0.0001
Guided alignment	-

Over the life of the project an increase of around 5 BLEU points was achieved through making changes to the model configuration. This illustrates the importance of not simply leaving the model in its default settings, but instead making as many improvements as possible.

5.2 Translation Engine Quality and Comparison

Figure 2 shows results for the optimal engine in comparison with the 3 publicly available en > cy engines of which we are aware. Google Translate and Microsoft Translator are well known API and browser-based services provided by multinational companies, while OPUS-MT-EN-CELTIC is an open-source transformer model available for download from the Hugging Face website.[9] All four are neural translation engines.

[9]https://huggingface.co/sshleifer/opus-mt-CELTIC-en. This model is trained on data from multiple Celtic languages and can translate English text to any of these languages. It was used instead of the en > cy model because its quality was higher at test time.

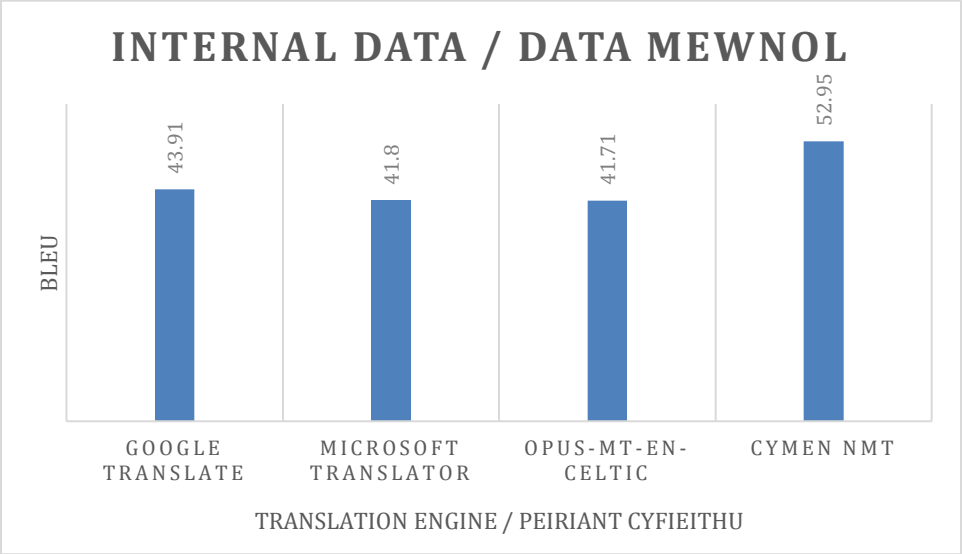

Figure 2 – Engine quality when translating Cymen's internal data

It is immediately apparent that Cymen's proprietary NMT engine translates the company's internal data to a far higher standard than the other services, performing around 9 BLEU better than the closest competitor, Google Translate. The most likely explanation for this is the nature of Cymen's training data. Cymen has access to a large amount of highly relevant data going back 20 years, including previous translations for clients who have submitted work consistently over that period. It is thus not particularly surprising that Cymen's engine performs better than the competition here, as the other institutions do not have access to the company's data.

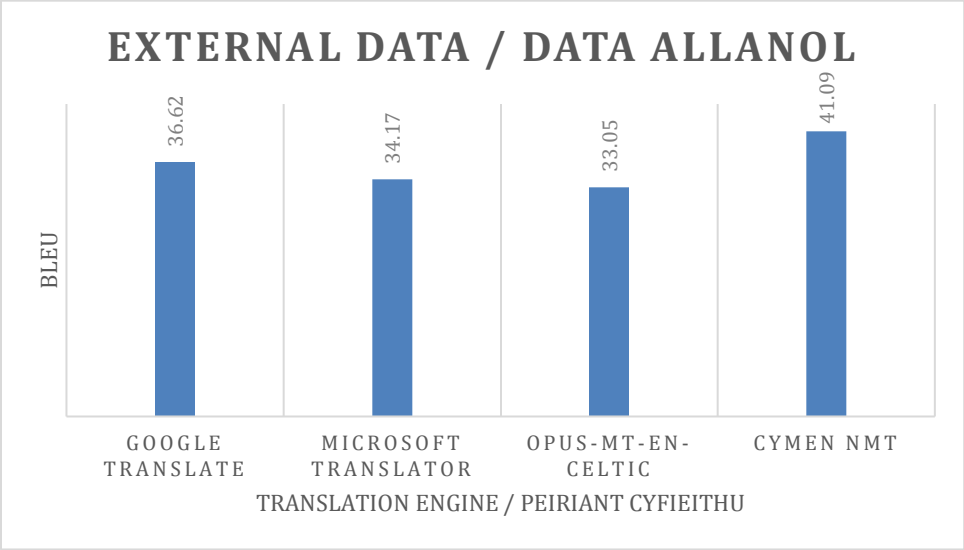

Figure 3 – Engine quality when translating external data

Figure 3 shows the quality of translations for the same translation engines for external data not belonging to Cymen. This test set was extracted from past translations carried out by Bangor University's Translation Unit. As a data set belonging to an institution with its own house style, it can be argued that this may be a fair test of the quality of these engines on 'neutral data'. Cymen's NMT engine once again performs ahead of the other engines here, although the difference is somewhat smaller – around 5 BLEU points this time.

What explains the superior performance of Cymen's NMT engine in this case? One possibility is that Cymen simply has more high-quality data for this language pair. The bulk of the training data is from Cymen's own translation memories, which constitute millions of words of professional, proofread translations. The possession of such an archive arguably represents a substantial advantage for the company over other institutions.

It is also worth noting the differences between the other translation engines. It appears that Google Translate was the second most successful in both experiments, scoring around 3 points above Microsoft Translator. OPUS-MT-EN-CELTIC and Microsoft Translator are relatively close – with around 1 BLEU point between them in both cases. This is an impressive result for the Hugging Face engine, given that the creators are presumably limited to publicly available training data.[10]

[10] The model's page lists Opus, Techiaith and BT (back translations) as data sets.

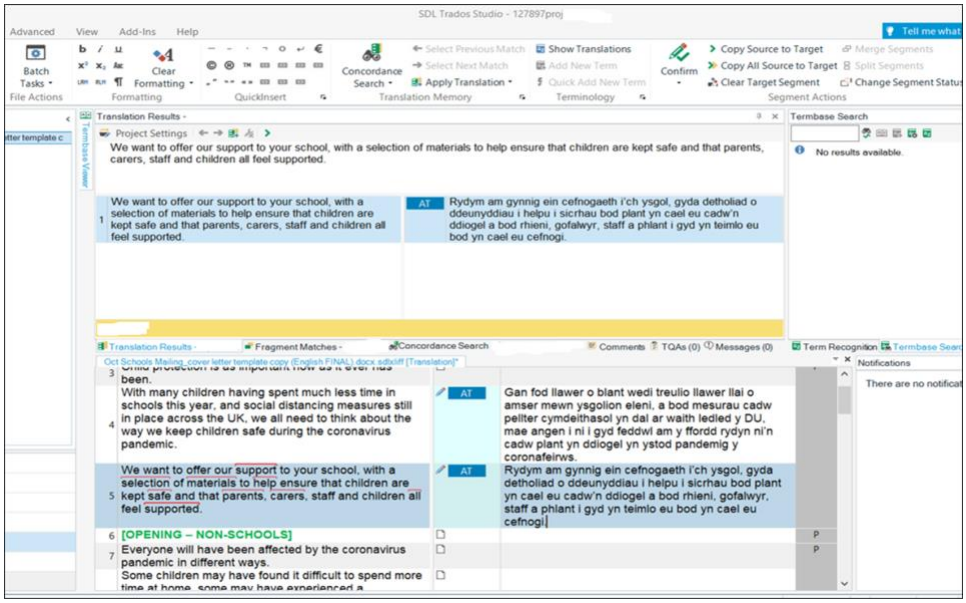

Figure 5 – Cymen's en > cy Marian engine running in Trados NMT

After optimizing the engines, the next step was to enable translators to use them. To do this, we adapted the plugin developed in the previous project to allow Marian engines to be accessed in Trados 2019. As in the previous project, we introduced the new engines gradually, but it quickly became clear that the translators' attitudes towards NMT were much more positive than towards the previous paradigm. Over time the company decided to use the technology in every single project by default, meaning that machine translation is now an integral part of the company's workflows.

5.3 Translator Feedback

To learn more about the translators' experiences of working with the translation engines, we decided to hold a series of focus groups involving the research officer (me), company management and translators. During these meetings, translators were able to discuss their positive and negative experiences of working with the translation engines. The feedback was generally highly positive, particularly as the project progressed and the translation engines were optimized. Many translators claimed that the translation engines allowed them to work more quickly and gave them more time for proofreading. Several translators claimed that using the engines allowed them to more than double the number of words that they were able to translate in a day.[11] As well as helping to speed up work, the translation engines could also be a psychological help for translators working on particularly long projects, due to the reduction in effort involved.

Among the most common negative comments was a tendency for the engines to incorrectly translate numbers and proper names. Once we were made aware of the problem, we were able to develop a simple workaround by making changes in the CAT tool used by our translators. Most translation programs now have QA features which

[11]Although this is of course anecdotal evidence at present, we hope to measure the perceived increase in productivity empirically soon.

allow translators to validate a document to ensure that numbers remain the same in both source and target segments, and that important terms are not translated incorrectly. All projects are now being created with these settings in place, thus protecting against this feature of the NMT engines.

Another problem reported in the focus group was the engines' inability to deal with XML markup while translating. XML tags often contain essential information for recreating the format of documents, so ensuring that tags are correctly placed around the relevant words in the target text is an all-important (but frustrating) part of a translator's daily work. SMT engines are able to deal with this relatively well, but it is considered a non-trivial problem for NMT. Strategies suggested have included injecting tags into parallel corpora for data augmentation [24], but for now our workaround involves stripping markup in the plugin and allowing translators to add the tags by hand.

Translators occasionally raised errors that were related more to the natural variation of register in the training data than any issues with the translation engines themselves. As a translation company, Cymen has its own house style which is normally adhered to in cases that do not contradict specific clients' own specified rules. In some cases, the translators objected to translations generated by the translation engine that violated either the house style or the translator's own preference. One example of this concerned translation of the English word 'whether' into Welsh. This is translated variously in the training data as *p'un ai a yw*, *p'un a yw*, *p'un ai yw* and more. After consulting with the translators, we decided to add automatic post-editing rules to the translation engine pipeline ensuring that the preferred form '*p'un a yw*' is always used in place of the other forms. Following the success of this approach we are now maintaining a file to which translators can add similar rules of their own directly. These kinds of measures may not be technologically sophisticated, but we feel that they are important in the sense that they increase translators' feeling of ownership over the technology, as well improving the quality of the output in a more general sense.

6 NEXT STEPS

There are many possible avenues to improving Cymen's translation engines further. The simplest way to do this would be to collect additional translation data. To some extent this will happen naturally in Cymen's case given that the company is currently adding around 1 million words a month to its translation memories. It would also be prudent to maintain an effort, along with other stakeholders, to pressure the Welsh Government to release further bilingual translation resources (TMXs, glossaries, and parallel corpora). Another method of collecting additional data is to use web crawlers. Bitextor[12] [25], for instance, is an open-source tool that can crawl parallel texts from bilingual websites, align segments to create a parallel corpus using Marian, and then clean the resulting corpus with the Bicleaner[13] tool. These resources have already been used with some success to increase the size of training data available for Irish (see section 2.3).

Another promising method is the creation of synthetic corpora. This technique is based around the fact that it is usually much easier to find monolingual data than bilingual data, particularly for domain specific material. First, a pre-trained translation engine is used to translate a monolingual data set to create a synthetic source side, in a process known as back-translation. The resulting synthetic corpus can then be combined with a normal parallel corpus. Researchers have reported substantial improvements from this technique, with Sennrich et al. [26]

[12] https://github.com/bitextor/bitextor

[13] https://github.com/bitextor/bicleaner

describing improvements of between 1 and 4 BLEU points for their models. This approach can also be combined with the Bitextor and Bicleaner tools discussed above to refine the data.

7 CONCLUSION

This chapter has described a simple project with far reaching benefits for a Welsh translation company. In contrast to the previous SMT technology, NMT translation engines were enthusiastically embraced by company translators, primarily due to their superior fluency, and they are now an integral part of Cymen's translation workflows. We have also demonstrated the advantages of using translation engines that are trained on Cymen's own archive of previous translations over services provided by third parties like Google and Microsoft. Not only are Cymen's bespoke engines empirically better at translating relevant data (see section 5.2), but they may also be a better option for machine translation of sensitive documents, and undoubtedly offer more flexibility in allowing the company to respond to translators' needs.

ACKNOWLEDGMENTS

This project was part funded by the Welsh Government through the SMART partnership scheme. I would like to thank Samantha Williams and Natalie Crawley from the Welsh Government for their tireless support over the last year. Thank you also to the staff at Cymen Cyf for their sanguinity during this and the previous KTP project, and for keeping an open mind throughout. Finally, I would like to thank Dewi Bryn Jones, Bangor University, for his hard work and patience as my project supervisor.

REFERENCES

[1] Welsh Government. 2019. 2050: A million Welsh speakers. Annual report 2017–18. Retrieved December 02, 2020 from https://gov.wales/sites/default/files/publications/2019-03/cymraeg-2050-a-millionwelsh-speakers-annual-report-2017-18.pdf

[2] Tegau Andrews. 2015. Cyd-destun gwleidyddol a chymdeithasol cyfieithu yn y Gymru gyfoes. In: Prys, D & Trefor, R. (Eds). Ysgrifau a Chanllawiau Cyfieithu. [Online]. Coleg Cymraeg Cenedlaethol. Carmarthen Retrieved on October 27, 2020 from https://llyfrgell.porth.ac.uk/media/ysgrifau-a-chanllawiau-cyfieithu-delythprys-arobat-trefor-goln

[3] Ben Screen. 2018. Defnyddio Cyfieithu Awtomatig a Chof Cyfieithu wrth gyfieithu o'r Saesneg i'r Gymraeg: Astudiaeth ystadegol o ymdrech, cynhyrchedd ac ansawdd gan ddefnyddio data Cofnodwyr Trawiadau Bysell a Thracio Llygaid. PhD Thesis, Department of Welsh, Cardiff University.

[4] Myfyr Prys and Dewi Bryn Jones. 2019. Embedding English to Welsh MT in a Private Company. In Proceedings of the Celtic Language Technology Workshop. European Association for Machine Translation, Dublin, 41-47.

[5] Antonio Lagarda, Vicent Alabau, Francisco Casacuberta, Roberto Silva and Enrique Díaz-de-Liaño. 2009. Statistical post-editing of a rule-based machine translation system. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers. Association for Computational Linguistics, Boulder, Colorado, USA, 217-220.

[6] Phillip Koehn. 2009. Statistical machine translation. Cambridge University Press.

[7] Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley and Andy Way. 2017. Is neural machine translation the new state of the art?. The Prague Bulletin of Mathematical Linguistics, 108, 1 (June 2017), 109-120.

[8] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers. Association for Computational Linguistics, Berlin, Germany, 131-198.

[9] Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, 257-267.

[10] Meghan Dowling, Teresa Lynn, Alberto Poncelas and Andy Way. 2018. SMT versus NMT: Preliminary Comparisons for Irish. In Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018). Association for Machine Translation in the Americas, Boston, USA, 12-20.

[11] Phillip Koehn. 2020. Neural machine translation. Cambridge University Press.

[12] Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, Philadelphia, USA, 311-318.

[13] Krzysztof Wołk and Krzysztof Marasek. 2015. Neural-based machine translation for medical text domain based on European medicines agency leaflet texts. Procedia Computer Science, 64 (October 2015), 2-9.

[14] Dimitar Shterionov, Riccardo Superbo, Pat Nagle, Laura Casanellas, Tony O'dowd and Andy Way. 2018. Human versus automatic quality evaluation of NMT and PBSMT. Machine Translation, 32, 3 (September 2018), 217-235.

[15] Francis Tyers and Kevin Donnelly. 2009. apertium-cy - a collaboratively-developed free RBMT system for Welsh to English. The Prague Bulletin of Mathematical Linguistics, 91 (January 2009), 57-66.

[16] Dafydd Jones and Andreas Eisele. 2006. Phrase-based statistical machine translation between English and Welsh. In Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages), LREC-2006. LREC, Genoa, Italy, 75-78.

[17] Meghan Dowling, Lauren Cassidy, Eimear Maguire, Teresa Lynn, Ankit Srivastava and John Judge. 2015. Tapadóir: Developing a statistical machine translation engine and associated resources for Irish. In Proceedings of the The Fourth LRL Workshop: "Language Technologies in support of Less-Resourced Languages". LRL, Poznan, Poland.

[18] Arne Defauw, Sara Szoc, Tom Vanallemeersch, Anna Bardadym, Joris Brabers, Frederic Everaert, Kim Scholte, Koen Van Winckel and Joachim Van den Bogaert. 2019. Developing a neural machine translation system for Irish. In Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages. European Association for Machine Translation, Dublin, Ireland, 32-38.

[19] Marta Bañón, Pinzhen (Patrick) Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Espla-Gomis, Mikel L. Forcada, et al.. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, 4555-4567.

[20] European Commission Representation in the UK, Chartered Institute of Linguists and the Institute of Translation and Interpreting. 2017. 2016 UK Translator Survey - Final Report. Retrieved September 28, 2020 from https://ec.europa.eu/unitedkingdom/sites/unitedkingdom/files/ukts2016-final-report-web_-_18_may_2017.pdf

[21] Gal Yarin and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. Advances in neural information processing systems, 29 (December 2016), 1019-1027.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems. NeuralIPS, Long Beach, CA, USA, 5998-6008.

[23] Jimmy Ba, Jamie Kiros and Geoffrey Hinton. 2016. Layer normalization. In Advances in NIPS 2016 Deep Learning Symposium. NeuralIPS, Barcelona, Spain, 1-14.

[24] Greg Hanneman and Georgiana Dinu. 2020. How Should Markup Tags Be Translated?. In Proceedings of the Fifth Conference on Machine Translation. Association for Computational Linguistics, Online, 1160-1173.

[25] Miquel Espla-Gomis. 2009. Bitextor, a free/open-source software to harvest translation memories from multilingual websites. In Proceedings of MT Summit XII, Association for Machine Translation in the Americas. Association for Machine Translation in the Americas, Ottawa, Canada.

[26] Rico Sennrich, Barry Haddow and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, 86-96.