applied sciences

MDPI

Article

Creating Welsh Language Word Embeddings

Padraig Corcoran [1], Geraint Palmer [2], Laura Arman [3], Dawn Knight [3] and Irena Spasić [1,*]

[1] School of Computer Science & Informatics, Cardiff University, Cardiff CF24 3AA, UK; CorcoranP@cardiff.ac.uk
[2] School of Mathematics, Cardiff University, Cardiff CF24 4AG, UK; palmergi1@cardiff.ac.uk
[3] School of English, Communication & Philosophy, Cardiff University, Cardiff CF10 3EU, UK; ArmanL@cardiff.ac.uk (L.A.); KnightD5@cardiff.ac.uk (D.K.)
* Correspondence: SpasicI@cardiff.ac.uk

Abstract: Word embeddings are representations of words in a vector space that models semantic relationships between words by means of distance and direction. In this study, we adapted two existing methods, word2vec and fastText, to automatically learn Welsh word embeddings taking into account syntactic and morphological idiosyncrasies of this language. These methods exploit the principles of distributional semantics and, therefore, require a large corpus to be trained on. However, Welsh is a minoritised language, hence significantly less Welsh language data are publicly available in comparison to English. Consequently, assembling a sufficiently large text corpus is not a straightforward endeavour. Nonetheless, we compiled a corpus of 92,963,671 words from 11 sources, which represents the largest corpus of Welsh. The relative complexity of Welsh punctuation made the tokenisation of this corpus relatively challenging as punctuation could not be used for boundary detection. We considered several tokenisation methods including one designed specifically for Welsh. To account for rich inflection, we used a method for learning word embeddings that is based on subwords and, therefore, can more effectively relate different surface forms during the training phase. We conducted both qualitative and quantitative evaluation of the resulting word embeddings, which outperformed previously described word embeddings in Welsh as part of larger study including 157 languages. Our study was the first to focus specifically on Welsh word embeddings.

Keywords: Welsh language; natural language processing; human language technology; machine learning; word embeddings

Citation: Corcoran, P.; Palmer, G.; Arman, L.; Knight, D.; Spasic, I. Creating Welsh Language Word Embeddings. *Appl. Sci.* **2021**, *11*, 6896. https://doi.org/10.3390/app11156896

Academic Editors: Rafael Valencia-Garcia and Francisco García-Sánchez

Received: 14 March 2021
Accepted: 21 July 2021
Published: 27 July 2021

1. Introduction

Natural language processing (NLP) studies the ways in which the analysis and synthesis of information expressed in a natural language can be automated. In recent years, most breakthroughs and improvements in the field have been the result of applying machine-learning techniques. One such case is that of word embeddings [1]. A word embedding is a mapping from the lexico-semantic space of words to the n-dimensional real valued vector space. Here, the dimensionality n is a hyper-parameter, i.e., a parameter whose value is set before the learning process begins. Compared to a traditional document–term matrix, whose second dimension will correspond to the size of the vocabulary, the dimension of the word embeddings is typically chosen to be relatively small, e.g., 300. Unlike document–term matrices, which are sparse, i.e., have a great many zero values, word embedding vectors are dense. The dimensions of word embedding vectors correspond to latent variables sampled from the distribution of words in a large corpus. As such, word embeddings tend to arrange semantically related words in similar spatial patterns. For example, the distance between the words 'shoe' and 'sock' should be relatively small compared to the distance between the words 'shoe' and 'butter'. Similarly, the vectors between 'foot' and 'sock' on one hand and 'hand' and 'glove' on the other should be near equal, i.e., have similar direction and magnitude. Owing to these latent semantic properties, it has

been demonstrated that in many cases the use of word embeddings improves performance of downstream NLP tasks such as named entity recognition and sentiment analysis.

To date, there has been much research on the creation of word embeddings for the English language [2]. In this study, however, we focus specifically on the Welsh language. Welsh is the native language of Wales, a country that is part of the United Kingdom (UK), in which it has the status of an official language alongside English. According to the 2011 UK Census, 19% of residents in Wales aged three and over were able to speak Welsh. Subsequently, the Office for National Statistics Annual Population Survey for the year ending in March 2019 determined that 896,900 Welsh residents (30% of the total population) aged three or over were able to speak Welsh. Nonetheless, Welsh is considered a low resource language in the sense that relative to English there are fewer corpora and NLP tools that are readily available. Empirical evidence suggests that the observance of lexico-semantic patterns in word embeddings is correlated with the size of corpus used for training [2].

Having assembled a large corpus of Welsh, the next challenge in training word embeddings is the recognition of words as discrete units of text, the process commonly known as tokenisation. The relative complexity of Welsh punctuation, particularly the extensive use of apostrophes that differs from their typical use in English, makes tokenisation challenging. Finally, Welsh is a morphologically rich language where inflection can give rise to multiple surface forms of a single word. Moreover, Welsh words can be inflected at their beginning as well as their ending, rendering the traditional stemming approaches ineffective in linking together related surface forms. Arguably, word embeddings trained on the original surface forms can capture the patterns of their inflection. However, such an approach is not feasible for languages that are highly inflected, yet low resourced as different surface forms may not occur frequently enough to establish a pattern of inflection.

Once these challenges are overcome, the actual process of training word embeddings is relatively straightforward as most of the state-of-the-art algorithms are, in fact, language independent. Several generic methods for learning word embeddings have been developed and applied successfully to different languages. However, such a general approach is not optimised with respect to the specific characteristics of individual languages and in turn the resulting word embeddings may not be optimal. In this study, we describe a novel workflow for training Welsh word embeddings that has been developed to overcome the above challenges. Specifically, we assembled a large corpus of Welsh. We considered several tokenisation methods including one designed specifically for Welsh. To account for word inflection, we opted for a generic word embedding method that is based on subwords and, therefore, can effectively relate different surface forms that share subwords.

The remainder of the paper is organised as follows. Section 2 presents a review of the state of the art in Welsh NLP. Section 3 presents the proposed workflow for learning Welsh word embeddings. Section 4 presents a qualitative and quantitative evaluation of the resulting word embeddings. The quantitative evaluation required us to create a Welsh word embedding benchmark. Finally, Section 5 draws conclusions from this study and discusses future research directions.

2. Related Work

This section reviews language resources that can support NLP in Welsh and, in particular, creation of word embeddings in this language. The first step in training word embeddings is to assemble a large corpus. Corpus-based language studies provide empirically based objective analyses of patterns of language as it is actually used, using evidence from a corpus (singular) or corpora (plural). CorCenCC (Corpws Cenedlaethol Cymraeg Cyfoes—the National Corpus of Contemporary Welsh) is a major corpus [3] containing 10 million words of written, spoken and digital (or 'e') Welsh language. It contains multiple language samples from real-life communication, allowing linguists to explore Welsh as it is actually used. It organises data into multiple facets, which can be used to study sublanguages as defined by [4]. All data are also annotated with different types of linguistic

information including morphological units, tokens, part-of-speech (POS) [5] and semantic categories [6,7]. In addition to linguistic research, the corpus can support a range of other applications such as learning and teaching of Welsh, but also NLP. In this study, together with other data sources, samples of data from an early release of CorCenCC were used to train word embeddings (see Section 3.1).

Having identified the relevant sources of data, the first step towards the creation of word embeddings is the process of identifying individual words as discrete units of text, the process known as tokenisation. Welsh Natural Language Toolkit (WNLT) [8] implements a set of rule-based Welsh NLP tools for tokenisation, lemmatisation, POS tagging and named entity recognition (NER), which are embedded into the GATE framework [9]. Similar NLP capabilities have been implemented to support pre-processing of documents stored in CorCenCC, which are tokenised and tagged using CyTag, a rule-based POS tagger [5]. 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) as well as the features specific to the Welsh language such as mutations [10]. The same team developed a lemmatiser, which can be used to normalise any inflected, mutated and/or conjugated word into its lemma [11].

Members of the CorCenCC team also developed downstream NLP methods for multi-word term recognition [12] and semantic tagging [6,7]. These methods were originally developed for English and successfully adapted for Welsh [13–15]. These methods can be useful for improving the performance of the downstream task of machine translation methods. For example, verbatim translations often deviate from the established terminology in the target language. Therefore, high-quality translations, performed by either humans or machines, require management of terminologies. Most machine translation systems require a terminology dictionary, e.g., [16,17] and/or the ability to extract terms dynamically [12] to support translations that use established terminology in the target language. In general, phrase-based statistical machine translation can improve the levels of translation quality where sufficiently large parallel corpora can be used for training as demonstrated in the case of English and Welsh [18]. In particular, the ability to align translated texts into paired sentences in the two languages [19] can support training of cross-lingual word embeddings [20], which can allow existing English language resources to be re-used for applications in Welsh.

Welsh word embeddings were first described in a study that presented a general method for creating word embeddings that was tested across 157 languages [2]. They were since used to support a machine-learning approach to a joint task of POS and semantic tagging [21]. However, these embeddings were created using a generic approach, which does not take into account specific characteristics of the Welsh language. For example, the text was segmented by the ICU tokeniser, which is language agnostic and not entirely appropriate for Welsh as it features an extensive use of apostrophes that differs from their typical use in other languages. Furthermore, a single method for creating word embeddings was considered, whereas alternative methods could prove to be more suitable for Welsh.

3. Methods

This section describes the proposed workflow for training word embeddings in Welsh. The workflow consists of three main steps. First, we assembled a large text corpus of Welsh language. Next, the corpus was pre-processed to identify individual words as discrete units of language. Finally, different methods for training word embeddings were applied to the pre-processed corpus. The following sections describe these steps in more detail.

3.1. Corpus Collection

Welsh is considered a low resource language in the sense that relative to English there are fewer corpora that are readily available. In particular, no single Welsh text corpus is large enough to train word embeddings. To support this goal specifically, we compiled a

large corpus of 92,963,671 words from 11 sources. Their summaries are provided in Table 1. Additional details are provided in the remainder of this section.

Table 1. Data sources used to collect the training corpus.

Source	Number of words
CorCenCC	1,875,540
Welsh Wikipedia	21,233,177
National Assembly for Wales 1999–2006	11,527,963
National Assembly for Wales 2007–2011	8,883,970
Cronfa Electroneg o Gymraeg	1,046,800
An Crúbadán	22,572,066
DECHE	2,126,153
BBC Cymru Fyw	14,791,835
Gwerddon	749,573
Welsh-medium websites	7,388,917
The Bible	749,573

- CorCenCC—CorCenCC is the first large-scale general corpus of Welsh language. The corpus currently contains over 10 million words of spoken, written and electronic language and collection is still ongoing. The corpus is designed to provide resources for the Welsh language that can be used in language technology (speech recognition, predictive text etc.), pedagogy, lexicography and academic research contexts among others. The development of CorCenCC was informed, from the outset, by representatives of all anticipated academic and community user groups. It therefore represents a user-driven model that will inform future corpus design, by providing a template for corpus development in any language and in particular lesser-used or minoritised languages. We obtained samples of some of the raw electronic text from an early release of the corpus, which included HTML web pages, and personal email and instant messaging correspondences, for use in the present study.
- Wikipedia—Wikipedia is a multilingual crowdsourced encyclopaedia. English version was the first edition of Wikipedia, which was founded in January 2001. As of 29 September 2019, it consists of 5,938,555 entries covering a wide range of subjects. Given its size and diversity, English Wikipedia is commonly used to train word embeddings in English. Welsh Wikipedia was founded in July 2003, but it is unfortunately still significantly smaller than its English counterpart. As of 29 September 2019, it consists of 106,128 entries.
- National Assembly for Wales 1999–2006—The National Assembly for Wales is the devolved parliament of Wales, which has many powers including those to make legislation and set taxes. The Welsh Language Act 1993 obliges all public sector bodies to give equal importance to both Welsh and English when delivering services to the public in Wales. This means that all documents shared by the National Assembly are available in both languages. By performing a web crawling, Jones er al. [18] assembled a parallel corpus from the public Proceedings of the Plenary Meetings of the Assembly between the years 1999–2006 inclusive. The authors used this corpus to support the development of a statistical machine translation method. For the purposes of our current study, we only used the Welsh language portion of the corpus.
- National Assembly for Wales 2007–2011—Similarly, Donnelly [22] created a parallel corpus from the same source but covering the period from 2007 until 2011. Again, we used the Welsh language portion of the corpus in the present study.
- Cronfa Electroneg o Gymraeg—This corpus consists of 500 articles of approximately 2000 words each, selected from a representative range of text types to illustrate modern (mainly post 1970) fiction and factual prose [23]. It includes articles from novels and short stories, religious writing, children literature, non-fiction material from education, science, business and leisure activities, public lectures, newspapers and magazines, reminiscences, academic writing, and general administrative materials.

- An Crúbadán—This corpus was created by [24] by crawling of Welsh text from Wikipedia, Twitter, blogs, the Universal Declaration of Human Rights and a Jehovah's Witnesses website (JW.org) [25]. To prevent duplication of data we removed all Wikipedia articles from this corpus before using it in the present study.
- DECHE—The Digitisation, E-publishing and Electronic Corpus (DECHE) project publishes e-versions of Welsh scholarly books that are out of print and unlikely to be re-printed in traditional paper format [26]. Books are nominated by lecturers working through the medium of Welsh and prioritised by the Coleg Cymraeg Cenedlaethol, which funds the project. We collected the text data from this project by downloading all e-books available.
- BBC Cymru Fyw—BBC Cymru Fyw is an online Welsh language service provided by BBC Wales containing news and magazine-style articles. Using the Corpus Crawler tool [27], we constructed a corpus containing all articles published on BBC Cymru Fyw between 1 January 2011 and 17 October 2019 inclusive.
- Gwerddon—Gwerddon is a Welsh-medium academic e-journal, which publishes research in arts, humanities and sciences. We downloaded all articles published in 29 editions of this journal.
- Welsh-medium websites—Golwg360 [28] and O'r Pedwar Gwynt [29] are Welsh-medium news websites. PoblCaerdydd [30] and Cylchgrawn Barn [31] are Welsh-medium online magazines.
- Beibl.net—The website beibl.net contains articles corresponding to all books of the Bible translated into an accessible variety of modern standard Welsh, along with informational pages.

3.2. Pre-Processing

Given a text corpus represented as a sequence of characters, tokenisation is the task of segmenting this sequence into tokens, which roughly correspond to words. Brute-force tokenisation, which removes punctuation and then identifies tokens as continuous character sequences between white spaces, oversimplifies the task and consequently achieves subpar results [32]. For example, consider the following sentence: "Mae'r haul yn taro'r paneli'n gyson â'n systemau'n rhedeg ar drydan wedi'i gynhyrchu yn y mis d'wetha' " (Engl. "The sun hits the panels consistently and our systems run on electricity produced in the last month"). The use of apostrophes in this example represents three different processes. For example, 'r is a form of the definite article which must follow a vowel (e.g., mae'r and taro'r). Similarly, 'n is either a function word yn reduced following a vowel (e.g., paneli'n and systemau'n) or a possessive pronominal, in this case ein (e.g., â'n systemau). Furthermore, 'i in wedi'i is another function word ei, an agreement proclitic following wedi. So far, these examples represent words or grammatical items separated from other words using an apostrophe. However, the last word d'wetha' represents a different use of apostrophe, which represents a less conservative written variety of standard Welsh by omitting the sounds of the full word diwethaf and shortening it to d'wetha' as is most commonly heard in standard speech. Clearly, one cannot assume that the apostrophe represents a word boundary.

We considered three tokenisation methods for the Welsh language. The first tokenisation method, used as a baseline, is a brute-force method consisting of the following steps. First, all characters are converted to lowercase. Next, all punctuation characters are removed. Finally, tokens are identified using white spaces. The second tokenisation method considered was the one from the Gensim library. This tokeniser returns tokens corresponding to maximal contiguous sequences of alphabetic characters. We chose not to remove accentuation from the lowercased text. The final tokeniser we considered was the one from the WNLT, which has been developed specifically for Welsh.

Following tokenisation, we removed rarely occurring tokens using by setting a threshold at 5 occurrences. This step helps to remove misspelled words from the corpus. No stemming or lemmatisation were performed because previous studies have found that

these actions remove information that can be used by machine learning to create language models [33].

3.3. Training

Traditionally, word embeddings are computed by minimising the distance between the words that appear in similar contexts. Prominent examples of such word embedding methods include word2vec [34], GloVe [35] and fastText [36]. Contextual word embeddings take this approach to the next level by creating different embeddings for the same word used in different contexts to convey different meanings. For example, the word 'bank' can be interpreted as either 'financial bank' and 'river bank' depending on the context. Examples of such word embedding methods include ELMo [37] and BERT [38]. Finally, word embeddings can be enriched with different types of information. For example, sentiment embeddings [39] incorporate the sentiment of words into their embeddings. The benefit of sentiment embeddings over standard embeddings is that the distance between opposite words such as 'good' and 'bad' that tend to appear in similar contexts will become larger to reflect their semantics more appropriately.

In this study, we focused solely on traditional word embedding methods, specifically word2vec [34] and fastText [36]. These approaches produce one vector per word, which enabled us to make direct comparison to the existing baseline approach [2]. Word2vec has two versions known as skip-gram and continuous bag of words (CBOW) respectively [34,40]. Given a target word w_t, skip-gram aims to predict its context. Formally, the objective of the skip-gram method is to maximize the log-likelihood defined in Equation (1), where $\mathcal{C}_t$ is the set of indices of context words surrounding the target word w_t.

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p(w_c|w_t) \tag{1}$$

For each word w, the skip-gram method defines two vectors u_w and v_w in $\mathbb{R}^n$, which are learnt automatically. These vectors are commonly referred to as input and output vectors respectively [36]. Given this, the skip-gram version estimates the probability $p(w_c|w_t)$ using the SoftMax function defined in Equation (2) where $s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is the scoring function defined in Equation (3) and $u_t^\top$ denotes the transpose of the vector u_t.

$$p(w_c|w_t) = \frac{\exp(s(w_t, w_c))}{\sum_{j=1}^{W} \exp(s(w_t, w_j))} \tag{2}$$

$$s(t, c) = u_t^\top v_c \tag{3}$$

This formulation of $p(w_c|w_t)$ renders the learning of the vectors u_w and v_w impractical because the cost of computing derivatives is proportional to the size of the corpus W. To overcome this challenge, [40] proposed two approximations known as hierarchical SoftMax and negative sampling.

Conversely, given a context, CBOW aims to predict the target word. Formally, instead of modelling $p(w_c|w_t)$, CBOW models $p(w_t|w_c)$ [34]. The fastText method generalises the two versions of word2vec, i.e., skip-gram and CBOW, by considering the subwords within the words [36]. The authors argue that this method is useful for morphologically rich languages such as Turkish and Finnish. Welsh is also morphologically rich, where inflection can give rise to multiple surface forms of a single word. Consider, for example, the word 'ci' (Engl. dog). In the phrase 'ei gi' (Engl. his dog), soft mutation applies to the word 'ci'. On the other hand, in the phrase 'ei chi' (Engl. her dog), aspirate mutation applies to the word 'ci' [41]. Therefore, both 'gi' and 'chi' correspond to the same lemma—'ci'. Mutations occur frequently in Welsh. Subword information has the potential to allow a word embedding method to relate different mutations of the same word. Therefore, fastText represents an appropriate choice of the word embedding method for this language.

FastText inserts special boundary characters < and > at the beginning and end respectively of each word. This allows the model to distinguish prefixes and suffixes from other character sequences. Each word w is then represented as a set $\mathcal{G}_w$ of character m-grams where the original word is also included in the set. For example, consider the case where w is the word 'sheep' and $m = 3$. In this case, {<sh, she, hee, eep, ep>, <sheep>}. For each m-gram g in the vocabulary, the fastText method defines a corresponding vector z_g, which is learnt. Given this, the fastText method is identical to the word2vec model except that the scoring function in Equation (3) is replaced by the scoring function in Equation (4).

$$s(t, c) = \sum_{g \in \mathcal{G}_t} z_g^\top v_c \tag{4}$$

In our experiments, the hyperparameters of the word2vec and fastText methods were set to the following values. For the fastText method we considered all m-grams for $3 \leq m \leq 6$. For both word2vec and fastText, we used a value of 300 for the dimension of the word embedding vectors n following the best practices described in [36] and trained for 20 epochs. We used the implementations of the word2vec and fastText methods available in the Gensim library.

4. Results and Analysis

Using different combinations of methods described in the previous section, we trained a total of 12 versions of Welsh word embeddings. We compared them against Welsh word embeddings described by [2,42] as the baseline.

Word embeddings are commonly evaluated using a combination of qualitative and quantitative methods. Qualitative methods involve manual selection of prototype words and inspection of their neighbourhood in the vector space. Quantitative methods can be divided into two categories, intrinsic and extrinsic methods [43,44]. Extrinsic methods evaluate word embeddings with respect to their effect on downstream NLP applications such as NER and sentiment analysis. Intrinsic methods evaluate how accurately word embeddings capture the semantic similarity of words under an assumption that semantically similar words will be close spatially in the vector space. In this study, we evaluated the word embeddings quantitatively using intrinsic methods. We considered four intrinsic methods, which are based on similarity, clustering, synonymy and analogy, respectively. Several the evaluation methods considered involved the creation of a corresponding dataset from English to Welsh. In all cases the translation in question was performed by a bilingual Welsh-English speaker. These translations were verified by a second bilingual Welsh-English speaker where any disagreement was resolved through discussion.

The remainder of the section provides further details of the experimental setup together with the corresponding results.

4.1. Word Similarity

Similarity-based methods for evaluating word embeddings use ground-truth pairwise semantic similarity of words, where semantic similarity is represented by a value within a fixed range with higher values indicating greater semantic similarity. The correlation of the semantic similarity of words and the cosine similarity of the corresponding word embeddings is used to gauge the utility of the word embeddings. The higher the correlation, the higher the utility of word embeddings. The ground-truth semantic similarity is typically estimated by native speakers.

We considered two word similarity datasets. First, the WordSimilarity-353 dataset contains a total of 353 word pairs in English [45,46]. Each word pair is associated with the mean taken from semantic similarity estimated independently by multiple individuals on a Likert scale from 0 to 10 inclusive. We adapted this dataset by translating it into Welsh using bilingual Welsh-English speakers. We used Spearman's rank correlation coefficient to measure the correlation between semantic similarity and cosine similarity. Table 2 provides the results. The naive tokenisation combined with CBOW performed best. In fact, all

CBOW methods regardless of tokenisation outperformed Grave's model. On the other hand, all methods based on skip-gram performed worse than Grave's model.

Table 2. The results achieved on the WordSimilarity-353 dataset.

Word Embedding Method	Tokenisation	Version	Semantic Correlation
fastText	naive	skip-gram	0.0495
fastText	naive	CBOW	0.1164
fastText	Gensim	skip-gram	0.0681
fastText	Gensim	CBOW	0.1326
fastText	WNLT	skip-gram	0.0632
fastText	WNLT	CBOW	0.1108
word2vec	naive	skip-gram	0.0448
word2vec	naive	CBOW	0.1157
word2vec	Gensim	skip-gram	0.0692
word2vec	Gensim	CBOW	0.1285
word2vec	WNLT	skip-gram	0.0604
word2vec	WNLT	CBOW	0.1067
Grave's	...	...	0.0785

The second dataset we considered was SimLex-999, which contains a total of 999 word pairs consisting of 666 noun pairs, 222 verb pairs and 111 adjective pairs in English [47,48]. Each word pair is associated with the mean semantic similarity estimated independently by 50 individuals on a Likert scale from 0 to 10 inclusive. We adapted this dataset by translating it into Welsh using bilingual Welsh-English speakers. Brute-force translation may introduce minor inaccuracies or other biases into the ground truth. For example, the word 'bank' has two interpretations, each being similar to the words 'river' and 'money' respectively. However, this homonymy is not observed in Welsh. Other considerations are cultural. For example, references to American concepts such as 'baseball', 'dollar' and 'buck' may occur rarely if at all in the Welsh corpus. Two word pairs with no Welsh equivalents, 'football–soccer' and 'dollar–buck', were removed from the dataset.

In addition to semantic similarity, SimLex-999 also provides the strength of free association represented as a value in the range 0 to 10 inclusive. For example, the words 'car' and 'petrol' are not semantically similar but have high free association. The strength of free association was calculated using the University of South Florida Free Association Dataset [49]. This dataset was generated by presenting human subjects with one of 5000 cue concepts and asking them to write the first word that comes to mind. Table 3 provides the results. The baseline performed the best, although all values are very low.

Table 3. The results achieved on the SimLex-999 dataset.

Word Embedding Method	Tokenisation	Version	Semantic Correlation	Free Association Correlation
fastText	naive	skip-gram	0.1131	0.0263
fastText	naive	CBOW	0.0692	0.0415
fastText	Gensim	skip-gram	0.1373	0.0471
fastText	Gensim	CBOW	0.0967	0.0466
fastText	WNLT	skip-gram	0.1246	0.0358
fastText	WNLT	CBOW	0.0941	0.0496
word2vec	naive	skip-gram	0.1075	0.0265
word2vec	naive	CBOW	0.0700	0.0427
word2vec	Gensim	skip-gram	0.1374	0.0461
word2vec	Gensim	CBOW	0.0975	0.0452
word2vec	WNLT	skip-gram	0.1247	0.0321
word2vec	WNLT	CBOW	0.0964	0.0491
Grave's	...	...	0.1466	0.0546

Note here that the Spearman correlation coefficients obtained in all cases were very low. This is expected, especially in the SimLex-999 dataset, with state-of-the-art English word embeddings, which were trained on corpus (around 1000 times larger than our Welsh corpus) also with very low correlations [47], between 0.2 and 0.45. There has also been discussion on the problems of using word similarity metrics such as these for evaluating word embeddings [50]. Thus, in the following sections we provide further alternative evaluation metrics.

4.2. Word Clustering

Concept categorisations is a common method for evaluating word embeddings [51]. It checks whether words can be grouped into natural categories from their vectors only. For example, the words 'bear' and 'bull' belong to an animal class, while cupboard and chair will belong to a furniture class.

We adapted a concept categorisation dataset from [52] by translating it into Welsh using bilingual Welsh-English speakers. It consists of 214 words assorted into 13 categories, which are provided in the Appendix A of this article. The vectors for each of the 214 words were clustered using k-means clustering combined with cosine distance and Euclidean distance respectively, with $k = 13$ to match the number of categories. Ideally, the 13 clusters should map directly to the 13 categories. Three measures were used to evaluate the clustering results.

- Purity measures the extent to which clusters contain words of the same category. It is calculated using Equation (5), where N is the number of words in total, M is the set of clusters and D is the set of known categories. It is calculated as the average count of the categories per cluster. Purity is commonly used to evaluate vector semantics, e.g., [51,53]. Its main shortcoming is that it does not penalise a single category being distributed over more than one cluster. For example, words belonging to an education category being distributed over more than one cluster.

$$P = \frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d| \tag{5}$$

- Rand Index measures the extent to which pairs of words that do or do not belong to the same category end up in the same cluster or not. For each word pair, clustering can produce a true positive (the same category and the same cluster), true negative (different categories and different clusters), false positive (different categories, but the same cluster), or false negative (the same category, but different clusters). The counts are given by TP, TN, FP and FN, respectively. These measures were used to measure accuracy in [52]. Rand index is calculated as the proportion of correctly predicted pairs as prescribed by Equation (6).

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

- Entropy measures how words from the same categories are distributed across the clusters. Low entropy indicates that words of the same category tend to be grouped within the same cluster. This measure, used in [53], is given in Equation (7).

$$E = -\sum_{m \in M} \frac{|m| \sum_{d \in D} \frac{|m \cap d|}{|m|} \ln\left(\frac{|m \cap d|}{|m|}\right)}{N \ln(|D|)} \tag{7}$$

Only the fastText models can be evaluated in this way. As word2vec models do not capture subword information, vectors for unseen words cannot be implied, and so the k-means algorithm fails. The results for the fastText models are given in Table 4.

WNLT tokenisation combined skip-gram performed best when Euclidean distance was used for clustering, while Gensim tokenisation combined with skip-gram performed

best when cosine distance was used for clustering. Grave's model performed the worst in both cases, indicating that a larger and more representative corpus yields word embeddings that perform better at concept categorisation tasks.

Table 4. The results for the concept categorisation task.

Word Embeddings			Euclidean Distance			Cosine Distance		
Method	**Tokenisation**	**Version**	**Purity**	**Rand Index**	**Entropy**	**Purity**	**Rand Index**	**Entropy**
fastText	naive	skip-gram	0.4860	0.8305	3.3298	0.6542	0.9178	2.2509
fastText	naive	CBOW	0.3832	0.7451	4.2892	0.5140	0.8826	3.4872
fastText	Gensim	skip-gram	0.5701	0.8811	2.8500	0.7242	0.9319	2.1092
fastText	Gensim	CBOW	0.4533	0.6958	3.9220	0.5140	0.8984	3.1224
fastText	WNLT	skip-gram	0.5794	0.8689	2.8448	0.6636	0.9225	2.2045
fastText	WNLT	CBOW	0.3972	0.7479	4.2486	0.5514	0.8692	3.2374
Grave's	...	...	0.2383	0.5640	5.3069	0.4486	0.7965	3.9540

4.3. Word Synonyms

The ability to link synonyms has been used to evaluate word embeddings [44,51] as well as other machine-learning tasks [54]. A dataset similar to the one based on multiple-choice synonym questions used in the Test of English as a Foreign Language was created for Welsh [55]. This is not a case of simple translation of English data, as synonymy is unique to a language and cannot be mapped easily from one language to another. Therefore, a new dataset was constructed with 50 questions (including nouns, adjectives, and verbs), given in the Appendix A of this article. Given a word (e.g., *rusty*) and a set of related words one of which is a (near)synonym (e.g., {*corroded*, *black*, *dirty*, *painted*}), an answer is selected as the word with the closest cosine distance. We measured the percentage of questions where the correct synonym (in this case *corroded*) was chosen.

Again, only the fastText models can be evaluated in this way, as those without subword information will be biased (will have fewer word choices) if presented with words not in the original dataset. The results for the fastText models are given in Table 5.

WNLT tokenisation combined with skip-gram performed the best on synonym prediction, while Grave's model performed the worst.

Table 5. The results achieved on the synonymy detection task.

Method	**Tokenisation**	**Version**	**% Correct**
fastText	naive	skip-gram	38%
fastText	naive	CBOW	32%
fastText	Gensim	skip-gram	36%
fastText	Gensim	CBOW	30%
fastText	WNLT	skip-gram	42%
fastText	WNLT	CBOW	30%
Grave's	...	...	28%

4.4. Word Analogies

Word analogies can be used to evaluate whether the semantic relationships between words correspond to the mathematical relationships between their respective embeddings. For example, the relationship between the word 'king' and 'queen' should be identical to the relationship between the words 'actor' and 'actress'. Therefore, if $\mathbf{x}_{\text{king}}$, $\mathbf{x}_{\text{queen}}$, $\mathbf{x}_{\text{actor}}$ and $\mathbf{x}_{\text{actress}}$ are the trained vectors for the words 'king', 'queen', 'actor' and 'actress', respectively, then we would expect:

$$\mathbf{x}_{\text{actor}} - \mathbf{x}_{\text{king}} + \mathbf{x}_{\text{queen}} \approx \mathbf{x}_{\text{actress}} \tag{8}$$

Given a set of examples of various language-specific relationships, the Gensim library allows us to measure the proportion for which the above equation holds, where the proximity is calculated using five nearest neighbours of the vector to the left. We

translated language-independent relationships such as those between nations and nationalities to Welsh. A dataset of grammatical relationships was constructed by a native Welsh speaking linguist and included adjective-opposite (435 pairs), adjective-comparative (55 pairs), adjective-superlative (55 pairs), adjectives-equative (55 pairs), nationalities (210 pairs), languages (120 pairs), noun-plural (6555 pairs), noun-singular (105 pairs), noun-gender (703 pairs), adjective-gender (105 pairs), adjective-plural (528 pairs), verb-nonfinite (325 pairs), verb-past-1st-singular (45 pairs), verb-past-3rd-singular (45 pairs), verb-past-impersonal (45 pairs), verb-present-impersonal (45 pairs), inflectional-preposition-1st-singular (36 pairs), and inflectional-preposition-2nd-plural (36 pairs).

The accuracy over all 9503 pairs, for each of the models, are given in Table 6. All models perform much better than Grave's model. There are also obvious rankings between tokenisation and training methods: gensim tokenisation performed better than WNLT, which in turn performed better than naive tokenisation; CBOW yields much more accurate models than skip-gram here; and, surprisingly, word2vec performs marginally better than fastText in all cases.

Table 6. The results achieved for the word analogy task.

Method	Tokenisation	Version	Accuracy
fastText	naive	skip-gram	16.64%
fastText	naive	CBOW	23.67%
fastText	Gensim	skip-gram	18.75%
fastText	Gensim	CBOW	28.01%
fastText	WNLT	skip-gram	17.97%
fastText	WNLT	CBOW	25.37%
word2vec	naive	skip-gram	16.66%
word2vec	naive	CBOW	23.88%
word2vec	Gensim	skip-gram	19.43%
word2vec	Gensim	CBOW	28.22%
word2vec	WNLT	skip-gram	18.21%
word2vec	WNLT	CBOW	25.21%
Grave's	...	...	9.00%

4.5. Qualitative Evaluation

We considered 30 nearest neighbours of a small set of prototype words. All prototype words were present in the text corpus used in training the word embeddings. Nearest neighbours were identified using the cosine similarity measure, which was calculated using the formula in Equation (9).

$$\text{similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \tag{9}$$

Given the cost of inspecting the neighbourhood manually, we limited this aspect of evaluation to comparison of a single model, namely the fastText skip-gram model with WNTL tokenisation, to the baseline. This model was chosen as it demonstrated the best results in our quantitative evaluation presented above. The following are a small selection of the words used for comparison and comments on the performance of the models:

- *nofio* (to swim): Both models list several sports and water-related activities as nearest neighbours although each model gives different words, including *beicio* (cycling), *caiacio* (kayaking), *syrffio* (surfing), *plymio* (diving), *nofwyr* (swimmers) and *pwll* (pool). Grave's model also lists several mutations and misspellings of *bwrcini* (burkini), which may indicate a small or un-diverse training corpus.
- *glaw* (rain): Our model lists a variety of weather phenomena including *eira* (snow), *gwyntoedd* (winds), *cawodydd* (showers), *cenllysg* (hail), *gwlyb* (wet), *stormydd* (storms), *corwyntoedd* (hurricanes), and *taranau* (thunder). Grave's model does list some related words such as *monswn* (monsoon), but mainly lists derivations of the original word

e.g., *glawiog* (rainy), and other unrelated words such as *car-boot* and *sgubai* (sweep), although this may relate to rain sweeping across the land.

- *hapus* (happy): Our model lists several synonyms or related adjectives, including *lwcus* (lucky), *falch* (glad), *ffodus* (fortunate), and *bodlon* (satisfied). Grave's model list some of these, but contains many other less similar words that could appear in the same context, such as *anhapus* (unhappy), *eisiau* (want), *teimlon* (felt), and *grac* (angry). It also lists words of similar spelling, but unrelated semantically: *siapus* (shapely) and *napus* (*brassica napus*; a species of rapeseed), which may indicate their model is relying too heavily on subword information.
- *meddalwedd* (software): Both models list many words related to computing and technology here, including *salwedd* (malware), *amgryptio* (encrypting), *cyfrifiadurol* (computational), *metaddata* (meta-data), *telegyfathrebu* (telecommunication), and *rhyngwyneb* (interface). Our model provides a greater variety of words, while Grave's model provides some English words and product names e.g., *DropBox*. This may be due to the fact that a more recent corpus was used by our model, as there will have been more technological articles published, and more technological terminology developed, in recent years.
- *ffrangeg* (the French language). There was a stark difference in the lists produced by the two models. Our model returned several other western European languages including *llydaweg* (Breton), *isalmaeneg* (Dutch), *galaweg* (Gallo) and *sbaeneg* (Spanish). Grave's model however gives several compound names, e.g., *Arabeg-Ffrangeg* (Arabic-French) and *FfrangegSaesneg* (French-English), while also returning several foreign words.
- *croissant* (the loan word 'croissant'): Again, there was a stark difference between the models here. Our model listed other foreign or loan words for food including *gefrüstuckt*, *brezel*, *müsli* and *spaghetti*, along with some unrelated foreign words. Grave's model lists several unrelated foreign words, many with similar spellings to the original word, e.g., *Eblouissant*, *Pourrissant*, and *Florissant*, again indicating the model's possible over-reliance on subword information.
- *gwario* (spend): Here Grave's model seems to group related words better. Our model lists mutations and conjugations of the original verb in addition to verbs formed by prefixation, e.g., *orwario* (overspend) and *tanwario* (underspend), and some other related words e.g., *wastraffu* (waste), *arbed* (save), and *talu* (pay). Grave's model lists other words related to finance including *Bitcoins*, *miliwnau* (millions), *arbedir* (to have saved), *chyllidebu* (budgeting) and *arian* (money).

An additional interesting observation was found by exploring the models manually. In our model, the nearest neighbours of some geographical places in Wales correspond to their geographical nearest neighbours:

- *Caerfyrddin*, a large town in West Wales has nearest neighbours made up from other towns in West Wales, for example *Llanelli*, *Aberteifi*, *Hwlffordd*, *Llambed*, *Penfro*, *Bwlchclawdd*, *Castellnewyddemlyn*, *Aberystwyth* and *Ceredigion*.
- *Caernarfon*, a large town in North Wales, has nearest neighbours made up from other towns in North Wales, for example *Dolgellau*, *Cricieth*, *Porthmadog*, *Llanllyfni*, *Pwllheli*, *Llangefni*, *Llandudno*, *Felinheli* and *Biwmares*.
- *Pontypridd*, a large town in the South Wales valleys, has nearest neighbours made up from other towns in the South Wales valleys, for example *Pontypŵl*, *Aberdâr*, *Pontyclun*, *Rhymni*, *Pontygwaith*, *Rhondda*, *Tonypandy*, *Abercynon* and *Trefforest*.

This pattern is not evident in Grave's model, which gives nearest neighbours to these towns most commonly as mutations and misspellings of the original word, and less commonly as Welsh towns further afield.

5. Conclusions

In this paper, we have presented a systematic evaluation of Welsh word embeddings trained using different combinations of word embedding and tokenisation approaches.

Although Welsh word embeddings have been created in the past, this is the first study that focuses solely on Welsh language and evaluates the embeddings with respect to its own patterns of syntax and semantics. In this respect, our model outperformed the only other existing model and as such sets the new baseline for the Welsh NLP community. To train the embeddings, we assembled the largest corpus of Welsh language. Although the corpus itself cannot be re-shared publicly due to data access restrictions, they can be collected from the original sources and used to re-create the corpus. Nonetheless, the word embeddings are made publicly available together with the associated code at [3].

Based on accuracy and consistency of performance on a wide variety of tasks, the recommendation arising from this study is to use fastText embeddings trained on WNLT-tokenised text using the skip-gram method. The only exception is word analogy for which word2vec embeddings trained on Gensim-tokenised text using the CBOW method performed better. These observations need to be taken into account when selecting the type of embeddings to support specific downstream tasks. For example, tasks such as document similarity may benefit from using word2vec embeddings, whereas tasks such as named entity recognition, which may require reasoning about newly encountered words, may benefit from using fastText embeddings.

In addition to this resource, we also created several datasets for the evaluation of word embeddings in Welsh. This study laid a foundation for developing cross-lingual word embeddings in which the vector space is shared between words in Welsh and English [56], where we demonstrated how NLP tools originally developed for English can be re-purposed for Welsh. Our future work will focus on learning contextual word embeddings in Welsh using approaches such BERT [57], where the same word can have a different vector representation depending on the current context. In addition, BERT generates embeddings at a subword level, which would help the out-of-vocabulary problem associated with small training datasets. In particular, using BERT to learn cross-lingual embeddings could effectively address the problem of code-switching and especially intra-word switching.

Author Contributions: Conceptualization, P.C., G.P., D.K. and I.S.; methodology, P.C., G.P., D.K. and I.S.; software, P.C., G.P., D.K. and I.S.; validation, P.C., G.P., L.A., D.K. and I.S.; formal analysis, P.C., G.P., L.A., D.K. and I.S.; investigation, P.C., G.P., D.K. and I.S.; resources, P.C., G.P., L.A., D.K. and I.S.; data curation, P.C., G.P., L.A., D.K. and I.S.; writing—original draft preparation, P.C., G.P., D.K. and I.S.; writing—review and editing, P.C., G.P., D.K. and I.S.; visualization, P.C., G.P., D.K. and I.S.; supervision, P.C., G.P., D.K. and I.S.; project administration, I.S.; funding acquisition, P.C., G.P., D.K. and I.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Welsh Government.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data and software to reproduce our results available at: https://datainnovation.cardiff.ac.uk/is/wecy/access.html. Accessed on 26 July 2021.

Acknowledgments: This work was funded by the Welsh Government. The authors would like to thank Gareth Morlais for his input on the creation of the datasets used to evaluate the Welsh language word embeddings.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Concept categorisation dataset.

Class	Words
anifail	arth, tarw, camel, cath, buwch, carw, ci, eleffant, ceffyl, cath bach, llew, mwnci, llygoden, wystrysen, ci bach, llygoden mawr, dafad, teigr, crwban, sebra
adeilad	ladd-dy, canolfan, clwb, dortur, tŷ gwydr, cyntedd, ysbyty, gwesty, tŷ, tafarn, llyfrgell, meithrin, bwyty, ysgol, nendwr, tafarndy, theatr, fila, puteindy
dillad	trôns, blows, cot, costiwm, megyg, het, siaced, jîns, mwclis, pyjamas, cochl, scarff, crys, siwt, trywsus, gwisg
crëwr	pensaer, arlunydd, adeiladwr, lluniwr, crefftwr, dylunydd, datblygwr, ffermwr, dyfeisiwr, crëwr, gwneuthurwr, cerddor, cychwynnwr, paentiwr, ffotograffydd, cynhyrchydd, teilwr
afiechyd	acne, anthracs, arthritis, asthma, canser, colera, sirosis, clefyd siwgwr, ecsema, ffliw, glawcoma, hepatitis, lewcemia, camfaethiad, llid yr ymennydd, pla, cryd cymalau, brech wen
teimlad	dicter, awydd, ofn, hapusrwydd, llawenydd, cariad, poen, angerdd, pleser, tristwch, sensitifrwydd, cywilydd, rhyfeddod
ffrwyth	afal, banana, aeron, ceirios, grawnwin, ciwi, lemwn, mango, melon, olewydd, oren, eirinen wlanog, gellygen, pinafal, mefys, dyfrfelon
dodrefn	gwely, silf llyfrau, cwpwrdd, cadair, cowtsh, crud, desg, tresel, lamp, lolfa, sedd, soffa, bwrdd, cwpwrdd dillad
corff	pigwrn, braich, clyst, llygad, gwyneb, bys, troed, llaw, pen, coes, trwyn, ysgwydd, byd troed, tafod, dant, addwrn
cyhoeddiad	atlas, llyfr, llyfryn, pamffled, catalog, llyfr coginio, geiriadur, gwyddoniadur, llawllyfr, cyfnodolyn, cylchgrawn, seinglawr, llyfr ffôn, cyfeirlyfr, gwerslyfr, llyfr gwaith
teulu	bachgen, plentyn, cefnithr, merch, tad, geneth, ŵyr, tadcu, nain, gŵr, crwt, mam, epil, sibling, mab, gwraig
amser	canrif, degawd, oed, noswaith, hydref, awr, mis, broe, nos, goramser, chwarter, tymor, semester, gwanwyn, haf, wythnos, penwythnos, gaeaf, blwyddyn
cerbyd	awyren, llong awyr, cerbyd, beic, cwch, car, criwser, hofrennydd, beic modur, tryc, roced, llong, lori, fan

Table A2. Synonym detection dataset.

Question	Synonym	Related Words
doeth	call	twp, ffôl, hapus
budr	brwnt	glân, gwyn, du
cyflym	clou	araf, hir, byr
bwrw	taro	mwytho, cyffwrdd, cicio
llefrith	llaeth	cwrw, caws, buwch
hawdd	rhwydd	anodd, arferol, rhydd
hynod	mor	tipyn, eithaf, bach
adnabyddus	enwog	dieithr, cerddor, amlwg
dolur	poen	moddion, salwch, meddyg
distaw	tawel	swnllyd, uchel, cyffroes
mân	bach	mawr, lled, cul
hogyn	bachgen	menyw, gwraig, myfyriwr
teyrngar	ffyddlon	celwydd, diafol, duw
merch	geneth	dyn, disgybl, gwr
blinedig	cysglyd	egnïol, gwelu, cysgu
creu	cynhyrchu	dinistrio, torri, adeiladu
rhyfel	brwydr	heddwch, byddin, milwr
rhwystro	atal	gadael, caniatáu, eisiau
edrych	sbio	clywed, methu, dweud
rhyfedd	anarferol	normal, lliwgar, doeth
anibendod	llanast	taclus, ystafell, brwnt
cnoi	brathu	bwyta, yfed, dannedd

Table A2. *Cont.*

Question	Synonym	Related Words
ceisio	trio	llwyddo, methu, cysgu
ffeindio	canfod	colli, nofio, chwilio
lleol	agos	pell, estron, gwyrdd
cwtogi	lleihau	ehangu, plygu, ymestyn
ehangu	tyfu	neidio, magu, meithrin
mur	wal	drws, ffenest, to
diogel	saff	peryglus, agored, rhwystredig
cyflawni	cwblhau	methu, gwagio, colli
deunydd	defnydd	dillad, pren, gwydr
gweiddi	bloeddio	sibrwd, siarad, peswch
pili pala	glöyn byw	pryfed, prycopyn, morgrugyn
teisen	cacen	bara, bisced, tost
digalon	trist	siriol, doniol, hwyl
cweryla	dadlau	cytuno, cusanu, bloeddio
hybu	hyrwyddo	digaloni, rhwystro, cynyddu
hardd	prydferth	hyll, esmwyth, caled
cymorth	help	rhwystr, moddion, athro
ffug	afreal	gwirioneddol, ffeithiol, stori
gwagle	gofod	llawn, awyr, bydysawd
bad	cwch	awyren, car, hofrennydd
dryll	gwn	cyllell, cleddyf, arf
cyfarfod	cwrdd	croesawi, siarad, ffarwelio
swydd	gwaith	gwyliau, rheswm, grym
holi	gofyn	ateb, gorymateb, meddwl
hanfodol	angenrheidiol	diangen, gwreiddiol, gwahanol
lleoliad	man	sefyllfa, amser, ystafell
creulon	cas	cwrtais, neis, crac
oherwydd	achos	rheswm, pam, esboniad

References

1. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
2. Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; Mikolov, T. Learning word vectors for 157 languages. In Proceedings of the International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018.
3. Corpws Cenedlaethol Cymraeg Cyfoes (Corcencc). 2020. Available online: https://github.com/CorCenCC (accessed on 26 July 2021).
4. Harris, Z. *A Theory of Language and Information: A Mathematical Approach*, 1st ed.; Clarendon Press: Oxford, UK, 1991.
5. Neale, S.; Donnelly, K.; Watkins, G.; Knight, D. 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, Miyazaki, Japan, 7–12 May 2018.
6. Piao, S.S.; Rayson, P.; Knight, D.; Watkins, G. Towards a Welsh semantic annotation system. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018; European Language Resources Association: Reykjavik, Iceland, 2018.
7. Piao, S.S.; Neale, S.; Ezeani, I.; Rayson, P.E.; Knight, D.; Donnelly, K. Open Welsh language resources for a corpus annotation framework. In Proceedings of the 10th International Corpus Linguistics Conference, Wales, UK, 23–27 July 2019.
8. Welsh Natural Language Toolkit (Wnlt). 2020. Available online: https://sourceforge.net/projects/wnlt-project/ (accessed on 26 July 2021).
9. Cunningham, H.; Maynard, D.; Bontcheva, K. *Text Processing with GATE (Version 6)*; University of Sheffield: Sheffield, UK, 2011.
10. Jones, D.B.; Robertson, P.; Prys, G Welsh Language Parts-of-Speech Tagger Api Service. 2015. Available online: http://techiaith.cymru/api/parts-of-speech-tagger-api/?lang=en (accessed on 26 July 2021).
11. Jones, D.B.; Robertson, P.; Prys, G. Welsh Language Lemmatizer Api Service. 2015. Available online: http://techiaith.cymru/api/lemmatizer/?lang=en (accessed on 26 July 2021).
12. Spasić, I.; Owen, D.; Knight, D.; Artemiou, A. Unsupervised multi–word term recognition in Welsh. In Proceedings of the Celtic Language Technology Workshop, Dublin, Ireland, 19 August 2019; pp. 1–6.
13. Spasić, I.; Greenwood, M.; Preece, A.; Francis, N.; Elwyn, G. Flexiterm: A flexible term recognition method. *J. Biomed. Semant.* **2013**, *4*, 27. [CrossRef] [PubMed]

14. Spasić, I. Acronyms as an integral part of multi-word term recognition—A token of appreciation. *IEEE Access* **2018**, *6*, 8351–8363. [CrossRef]
15. Rayson, P.; Archer, D.; Piao, S.; McEneryb, T. The UCREL semantic analysis system. In Proceedings of the LREC-04 Workshop, Beyond Named Entity Recognition Semantic Labelling for NLP, Lisbon, Portugal, 25 May 2004; pp. 7–12.
16. Welsh National Language Technologies Portal. Terminology Dictionary Widget. 2020. Available online: http://techiaith.cymru/cloud/widgets/terminology-dictionary-widget/?lang=en (accessed on 26 July 2021).
17. Welsh Government. Termcymru. 2020. Available online: https://gov.wales/bydtermcymru (accessed on 26 July 2021).
18. Jones, D.; Eisele, A. 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, Genoa, Italy, 24–26 May 2006.
19. Welsh National Language Technologies Portal. Welsh-English Aligner. 2020. Available online: http://techiaith.cymru/translation/aligner/?lang=en (accessed on 26 July 2021).
20. Ruder, S.; Vulić, I.; Søgaard, A. A survey of cross-lingual word embedding models. *J. Artif. Intell. Res.* **2019**, *65*, 569–631. [CrossRef]
21. Ezeani, I.; Piao, S.S.; Neale, S.; Rayson, P.; Knight, D. Leveraging pre-trained embeddings for Welsh taggers. In Proceedings of the 4th Workshop on Representation Learning for NLP, Florence, Italy, 2 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 270–280.
22. Donnelly, K. Kynulliad3: A Corpus of 350,000 Aligned Welsh-English Sentences from the Third Assembly (2007–2011) of the National Assembly for Wales. 2013. Available online: http://cymraeg.org.uk/kynulliad3 (accessed on 26 July 2021).
23. Ellis, , N.C.; O'Dochartaigh, C.; Hicks, W.; Morgan, M.; Laporte, N. Cronfa Electroneg o Gymraeg (ceg): A 1 Million Word Lexical Database and Frequency Count for Welsh. 2001. Available online: https://www.bangor.ac.uk/canolfanbedwyr/ceg.php.en (accessed on 26 July 2021).
24. Scannell, K.P. The crúbadán project: Corpus building for under-resourced languages. In Proceedings of the 3rd Web as Corpus Workshop, Louvain-la-Neuve, Belgium, 15–16 September 2007.
25. Tystion Jehofa. 2020. Available online: https://www.jw.org/cy/ (accessed on 26 July 2021).
26. Prys, D.; Jones, D.; Roberts, M. Deche and the Welsh national corpus portal. In Proceedings of the First Celtic Language Technology Workshop, Dublin, Ireland, 23 August 2014; pp. 71–75.
27. Corpus Crawler. 2020. Available online: https://github.com/google/corpuscrawler (accessed on 26 July 2021).
28. golwg360. 2020. Available online: https://golwg360.cymru (accessed on 26 July 2021).
29. O'r Pedwar Gwynt. 2020. Available online: https://pedwargwynt.cymru (accessed on 26 July 2021).
30. Pobl Caerdydd. 2020. Available online: https://poblcaerdydd.com/ (accessed on 26 July 2021).
31. Cylchgrawn Barn. 2020. Available online: https://barn.cymru/ (accessed on 26 July 2021).
32. Manning, C.D.; Raghavan, P.; Schütze, H. Introduction to information retrieval. In *An Introduction To Information Retrieval*; Cambridge University Press: Cambridge, MA, USA, 2008; Volume 151, p. 5.
33. Howard, J. Lesson 12. In *Practical Deep Learning for Coders*; fast.ai. Available online: https://course.fast.ai/ (accessed on 26 July 2021).
34. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
35. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
36. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
37. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 21–23 June 2018; pp. 2227–2237.
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
39. Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; Zhou, M. Sentiment embeddings with applications to sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 496–509. [CrossRef]
40. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
41. King, G. *Modern Welsh: A Comprehensive Grammar*; Routledge: London, UK, 2015.
42. Fasttext Word Vectors. 2020. Available online: https://fasttext.cc/docs/en/crawl-vectors.html (accessed on 26 July 2021).
43. Schnabel, T.; Labutov, I.; Mimno, D.; Joachims, T. Evaluation methods for unsupervised word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 298–307.
44. Bakarov, A. A survey of word embeddings evaluation methods. *arXiv* **2018**, arXiv:1801.09536.

45. Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; Ruppin, E. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.* **2002**, *20*, 116–131.
46. The Wordsimilarity-353 Test Collection. Available online: http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/ (accessed on 26 July 2021).
47. Hill, F.; Reichart, R.; Korhonen, A. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Comput. Linguist.* **2015**, *41*, 665–695. [CrossRef]
48. Simlex-999. 2020. Available online: https://fh295.github.io/simlex.html (accessed on 26 July 2021).
49. Nelson, D.L.; McEvoy, C.L.; Schreiber, T.A. The university of south florida free association, rhyme, and word fragment norms. *Behav. Res. Methods Instrum. Comput.* **2004**, *36*, 402–407. [CrossRef] [PubMed]
50. Faruqui, M.; Tsvetkov, Y.; Rastogi, P.; Dyer, C. Problems with evaluation of word embeddings using word similarity tasks. *arXiv* **2016**, arXiv:1605.02276.
51. Baroni, M.; Dinu, G.; Kruszewski, G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 238–247.
52. Almuhareb, A.; Poesio, M. Attribute-based and value-based clustering: An evaluation. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 158–165.
53. Almuhareb, A.; Poesio, M. Concept learning and categorization from the web. In Proceedings of the Annual Meeting of the Cognitive Science Society, Stresa, Italy, 21–23 July 2005; Volume 27.
54. Turney, P.D. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 491–502.
55. Landauer, T.K.; Dumais, S.T. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* **1997**, *4*, 211–240. [CrossRef]
56. Espinosa-Anke, L.; Palmer, G.; Corcoran, P.; Filimonov, M.; Spasić, I.; Knight, D. English-Welsh cross-lingual embeddings. *Appl. Sci.* **2021**, in press. [CrossRef]
57. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, MN, USA, 2–9 June 2019; pp. 4171–4186.