Igbo Diacritic Restoration using Embedding Models

Ignatius Ezeani **Mark Hepple** **Ikechukwu Onyenwe** **Chioma Enemuo**
Department of Computer Science,
The University of Sheffield, United Kingdom.
`https://www.sheffield.ac.uk/dcs`
`{ignatius.ezeani, m.r.hepple, i.onyenwe, clenemuo1}@sheffield.ac.uk`

Abstract

Igbo is a low-resource language spoken by approximately 30 million people world-wide. It is the native language of the Igbo people of south-eastern Nigeria. In Igbo language, diacritics - orthographic and tonal - play a huge role in the distinction of the meaning and pronunciation of words. Omitting diacritics in texts often leads to lexical ambiguity. Diacritic restoration is a pre-processing task that replaces missing diacritics on words from which they have been removed. In this work, we applied embedding models to the diacritic restoration task and compared their performances to those of *n*-gram models. Although word embedding models have been successfully applied to various NLP tasks, it has not been used, to our knowledge, for diacritic restoration. Two classes of word embeddings models were used: those projected from the English embedding space; and those trained with Igbo bible corpus (≈ 1m). Our best result, 82.49%, is an improvement on the baseline *n*-gram models.

1 Introduction

Lexical disambiguation is at the heart of a variety of NLP tasks and systems, ranging from grammar and spelling checkers to machine translation systems. In Igbo language, diacritics - orthographic and tonal - play a huge role in the distinction of the meaning and pronunciation of words (Ezeani et al., 2017, 2016). Therefore, effective restoration of diacritics not only improves the quality of corpora for training NLP systems but often improves the performance of existing ones (De Pauw et al., 2007; Mihalcea, 2002).

1.1 Diacritic Ambiguities in Igbo

There is a wide range of ambiguity classes in Igbo (Thecla-Obiora, 2012). In this paper, we focus on diacritic ambiguities. Besides orthographic diacritics (i.e. dots below and above), tone marks also impose the actual pronunciation and meaning on different words with the same latinized spelling. Table 1 shows Igbo diacritic complexity which impacts on word meanings and pronunciations[1].

Char	**Ortho**	**Tonal**
a	–	à,á, ā
e	–	è,é, ē
i	ị	ì, í, ī, ị̀, ị́, ị̄
o	ọ	ò, ó, ō, ọ̀, ọ́, ọ̄
u	ụ	ù, ú, ū, ụ̀, ụ́, ụ̄
m	–	m̀,ḿ, m̄
n	ṅ	ǹ,ń, n̄

Table 1: Igbo diacritic complexity

An example of lexical ambiguity caused by the absence of tonal diacritics is the word *akwa* which could mean **ákwá** (cry), **àkwà** (bed/bridge), **ákwà** (cloth) and **àkwá** (egg). Another example of ambiguity due to lack of orthographic diacritics is the word *ugbo* which could mean **ụ́gbọ́** (craft:car|boat|plane); **úgbō** (farm).

1.2 Proposed Approach

As shown in section 2, previous approaches to diacritic restoration techniques depend mostly on existing human annotated resources (e.g. POS-tagged corpora, lexicon, morphological information). In this work, embedding models were used to restore diacritics in Igbo. For our experiments, models are created by training or projection. The evaluation method is a simple accuracy measure i.e. the average percentage of correctly restored instances over all instance keys. An accuracy of

[1] In Igbo, *m* and *n* are *nasal consonants* which are in some cases treated as tone marked vowels.

Proceedings of NAACL-HLT 2018: Student Research Workshop, pages 54–60
New Orleans, Louisiana, June 2 - 4, 2018. ©2017 Association for Computational Linguistics

82.49% is achieved with the **IgboBible** model using **Tweak3** confirming our hypothesis that the semantic relationships captured in embedding models could be exploited in the restoration of diacritics.

2 Related Works

Some of the key studies in diacritic restoration involve word-, grapheme-, and tag-based techniques (Francom and Hulden, 2013). Some examples of word-based approaches are the works of Yarowsky (Yarowsky, 1994., 1999) which combined decision list with morphological and collocational information.

Grapheme-based models tend to support low resource languages better by using character collocations. Mihalcea *et al* (2002) proposed an approach that used character based instances with classification algorithms for Romanian. This later inspired the works of Wagacha *et al* (2006), De Pauw *et al* (2011) and Scannell (2011) on a variety of relatively low resourced languages. However, it is a common position that the word-based approach is superior to character-based approach for well resourced languages.

POS-tags and language models have also been applied by Simard (1998) to well resourced languages (French and Spanish) which generally involved *pre-processing*, *candidate generation* and *disambiguation*. Hybrid techniques are common with this task e.g. Yarowsky (1999) used decision list, Bayesian classification and Viterbi decoding while Crandall (2005) applied Bayesian- and HMM-based methods. Tufiş and Chiţu (1999) used a hybrid approach that backs off to character-based method when dealing with "unknown words".

Electronic dictionaries, where available, often augment the *substitution schemes* used (Šantić et al., 2009). On Maori, Cocks and Keegan (2011) used naïve Bayes algorithms with word n-grams to improve on the character based approach by Scannell (2011).

For Igbo, however, one major challenge to applying most of the techniques mentioned above that depend on annotated datasets is the lack of these datasets for Igbo e.g tagged corpora, morphologically segmented corpora or dictionaries. This work aims at using a resource-light approach that is based on a more generalisable state-of-the-art representation model like word-embeddings which could be tested on other tasks.

2.1 Igbo Diacritic Restoration

Igbo was among the languages in a previous work (Scannell, 2011) with 89.5% accuracy on web-crawled Igbo data (31k tokens with a vocabulary size of 4.3k). Their *lexicon lookup* methods, *LL* and *LL2* used the most frequent word and a bigram model to determine the right replacement. However, their training corpus was too little to be representative and there was no language speaker in their team to validate their results.

Ezeani *et al* (2016) implemented a more complex set of n–gram models with similar techniques on a larger corpus and reported better results but their evaluation method assumed a closed-world by training and testing on the same dataset. Better results were achieved with the approach reported in (Ezeani et al., 2017) but it used a non-standard data representation model which assigns a sequence of real values to the words in the vocabulary. This method is not only inefficient but does not capture any relationship that may exist between words in the vocabulary.

Also, for Igbo, diacritic restoration does not always eliminate the need for sense disambiguation. For example, the restored word *àkwà* could be referring to either *bed* or *bridge*. Ezeani *et al* (2017) had earlier shown that with proper diacritics on ambiguous wordkeys[2](e.g. *akwa*), a translation system like *Google Translate* may perform better at translating Igbo sentences to other languages. This strategy, therefore, could be more easily extended to sense disambiguation in future.

Statement	***Google Translate***	**Comment**
O ji *egbe* ya gbuo *egbe*	He used his **gun** to kill *gun*	wrong
O ji **égbè** ya gbuo **égbé**	He used his **gun** to kill **kite**	correct
Akwa ya di n'elu *akwa* ya	It was on the **bed** in his room	fair
Ákwà ya di n'elu **àkwà** ya	his **clothes** on his **bed**	correct
Oke riri *oke* ya	Her addiction	confused
Òké riri **òkè** ya	**Mouse** ate his **share**	correct
O jiri *ugbo* ya bia	He came with his *farm*	wrong
O jiri **ụgbọ** ya bia	He came with his **car**	correct

Table 2: Disambiguation challenge for *Google Translate*

[2] A *wordkey* is a "latinized" form of a word i.e. a word stripped of its diacritics if it has any. Wordkeys could have multiple diacritic variants, one of which could be the same as the wordkey itself.

3 Embedding Projection

Embedding models are very generalisable and therefore will be a good resource for Igbo which has limited resources. We intend to use both trained and projected embeddings for the task. The intuition for embedding projection, illustrated in Figure 1, is hinged on the concept of the universality of meaning and representation.

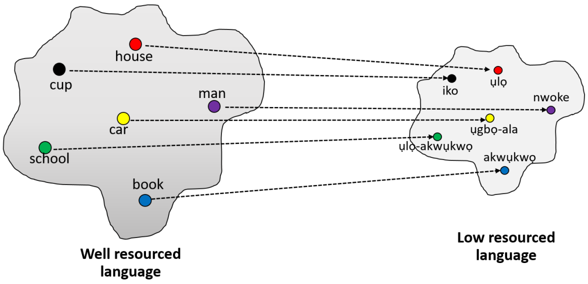

Figure 1: Embedding Projection

We adopt an alignment-based projection method similar to the one described in (Guo et al., 2015). It uses an Igbo-English alignment dictionary $A^{I|E}$ with a function $\boldsymbol{f}(w_i^I)$ that maps each Igbo word w_i^I to all its co-aligned English words $w_{i,j}^E$ and their counts $c_{i,j}$ as defined in Equation 1. $|V^I|$ is the vocabulary size of Igbo and n is the number of co-aligned English words.

$$\begin{aligned} A^{I|E} &= \{w_i^I, \boldsymbol{f}(w_i^I)\}; i = 1..|V^I| \\ \boldsymbol{f}(w_i^I) &= \{w_{i,j}^E, c_{i,j}\}; j = 1..n \end{aligned} \tag{1}$$

The projection is formalised as assigning the weighted average of the embeddings of the co-aligned English words $w_{i,j}^E$ to the Igbo word embeddings $\mathbf{vec}(w_i^I)$ (Guo et al., 2015):

$$\mathbf{vec}(w_i^I) \leftarrow \frac{1}{C} \sum_{w_{i,j}^E), c_{i,j} \in f(w_i^I)} vec(w_{i,j}^E) \cdot c_{i,j} \tag{2}$$

where $C \leftarrow \sum_{c_{i,j} \in f(w_i^I)} c_{i,j}$

4 Experimental Setup

4.1 Experimental Data

We used the English-Igbo parallel bible corpora, available from the *Jehova Witness* website[3], for our experiments. The basic statistics are presented in Table 3[4].

Item	Igbo	English
Lines	32416	32416
Words+puncs	1,070,708	1,048,268
Words only	902,429	881,771
Unique words	16,084	15,000
Diacritized words	595,221	–
Unique diacritized words	8,750	–
All wordkeys	15,476	–
Unique wordkeys	14,926	–
Ambiguous wordkeys:	550	
– 2 variants	516	–
– 3 variants	19	–
– 4 variants	9	–
– 5 variants	3	–
– 6 variants	3	–

Table 3: Corpus statistics

Table 3 shows that both the total corpus words and its word types constitute over 50% diacritic words i.e. words with at least one diacritic character. Over 97% of the ambiguous wordkeys have 2 or 3 variants.

4.2 Experimental Datasets

We chose 29 wordkeys which have several variants occurring in our corpus, the wordkey itself occurring too[5]. For each wordkey, we keep a list of sentences (excluding punctuations and numbers), each with a blank (see Table 5) to be filled with the correct variant of the wordkey.

4.3 Experimental Procedure

The experimental pipeline, as illustrated in Figure 2, follows three fundamental stages:

4.3.1 Creating embedding model

Four embedding models, two trained and two projected, were created for Igbo in the first stage of the pipeline:

Trained: The first model, **IgboBible**, is produced from the data described in Table 3 using the Gensim *word2vec* Python libraries (Řehůřek and Sojka, 2010). Default

[3] jw.org

[4] In these counts, the case of words is preserved e.g. *Ọtụtụ* and *ọtụtụ* have different counts

[5] Highly dominant variants or very rarely occurring wordkeys were generally excluded from the datasets.

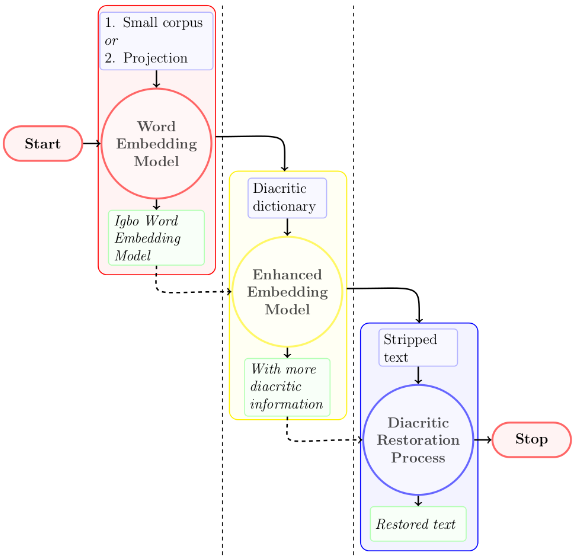

Figure 2: Pipeline for Igbo Diacritic Restoration using Word Embedding

configurations were used apart from optimizing $dimension(default = 100)$ and $window\_size(default = 5)$ parameters to 140 and 2 respectively on the **Basic** restoration method described in section 4.3.3.

We also used **IgboWiki**, a pre-trained Igbo model from *fastText Wiki word vectors*[6] project (Bojanowski et al., 2016).

Projected Using the projection method defined above, we created the **IgboGNews** model from the pre-trained *Google News*[7]*word2vec* model while the **IgboEnBbl** is projected from a model we trained on the English bible.

Table 4 shows the vocabulary sizes ($\#|V|^L$) for embedding models of each language L, as well as the dimensions (**#vecs**) of each of the models used in our experiments. While the pre-trained models and their projections have vector sizes of 300, our trained **IgboBible** performed best with vector size of 140 and so we trained the **IgboEnBbl** with the same dimension.

Model	$\#\lvert V\rvert^I$	**#vecs**	$\#\lvert V\rvert^E$	**#data**
IgboBible	4968	140	–	902.5k
IgboWiki	3111	300	–	(unknown)
IgboGNews	3046	300	3m	100bn
IgboEnBbl	4057	140	6.3k	881.8k

Table 4: Igbo and English models: vocabulary, vector and training data sizes

IgboGNews has a lot of *holes* i.e. 1101 out of 4057, (24.92%) entries in the alignment dictionary words were not represented in the Google News embedding model. A quick look at the list revealed that they are mostly bible names that do not exist in the Google News model and so have no vectors for their Igbo equivalents e.g. *kọrịnt, nimshaị, manase, peletaịt, gọg, pileg, abịshag, arọna, frankịnsens*.

The projection process removes[8] these words thereby stripping the model of a quarter of its vocabulary with any linguistic information from them.

4.3.2 Deriving *diacritic* embedding models

In both training and projection of the embedding model, vectors are assigned to each word in the dictionary, and that includes each diacritic variant of a wordkey. The **Basic** restoration process (*section* 4.3.3) uses this initial embedding model *as-is*. The models are then refined by "tweaking" the variant vectors to get new ones that correlate more with context embeddings.

For example, let mcw_v contain the top n of the most co-occurring words of a certain variant, v and their counts, c. The following three *tweaking* methods are applied:

- **Tweak1**: adds to each diacritic variant vector the weighted average of the vectors of its most co-occurring words (see Equation (3)). At restoration time, *all* the words in the sentence are used to build the context vector.

- **Tweak2**: updates each variant vector as in *Tweak1* but its restoration process uses *only* the vectors of co-occurring words with each of the contesting variants excluding common words.

- **Tweak3**: is similar to the previous methods but *replaces* (not *updates*) each of the variant

[6] Pre-trained on 294 different languages of Wikipedia

[7] https://code.google.com/archive/p/word2vec/

[8] Other variants of this process assign zero vectors to these words or remove the same words from the other models.

Variant	Left context	Placeholder	Right context	Meaning
àkwá	ka okwa nke kpokotara	____	o na-eyighi eyi otu	egg
ákwà	a kpara akpa mee	____	ngebichi nke onye na-ekwe	cloth
ákwá	ozugbo m nuru mkpu	____	ha na ihe ndi a	cry

Table 5: Instances of the wordkey ***akwa*** in context

vectors (see Equation (4)).

$$\mathbf{diac_{vec}} \leftarrow diac_{vec} + \frac{1}{|mcw_v|} \sum_{w \in mcw_v} w_{vec} * w_c \quad (3)$$

$$\mathbf{diac_{vec}} \leftarrow \frac{1}{|mcw_v|} \sum_{w \in mcw_v} w_{vec} * w_c \quad (4)$$

where w_c is the 'weight' of w i.e. the probability distribution of the count of w in mcw_v.

4.3.3 Diacritic restoration process

Algorithm 1 sketches the steps followed to apply the diacritic embedding vectors to the diacritic restoration task. This algorithm is based on the assumption that combining the vectors of words in context is likely to yield a vector that is more similar to the correct diacritic variant. In this process, a set of candidate vectors, $D^{wk} = \{d_1, ..., d_n\}$ for each wordkey, wk, are extracted from the embedding model. C is defined as the list of the context words of a sentence containing a placeholder (examples are shown in Table 5) to be filled and vec_C is the context vector of C (Equation (5)).

Algorithm 1 Diacritic Restoration Process

```
Require: Embedding & instances with blanks
Ensure: Blanks filled with variants
load embeddings and instances
for each instance do
    Get candidate vectors:$D^{wk}$
    $\mathbf{vec_C} \leftarrow \frac{1}{|C|} \sum_{w \in C} embed[w]$ (5)
    $\mathbf{diac_{best}} \leftarrow \mathbf{argmax}_{d_i \in D^{wk}} sim(\mathbf{vec_C}, d_i)$ (6)
end for
```

5 Evaluation Strategies

A major subtask of this project is building the dataset for training the embedding and other language models. For all of the 29 wordkeys[9] used in the project, we extracted 38,911 instances each with the correct variant and no diacritics on all words in context. The dataset was used to optimise the parameters in the training of the **Basic** embedding model. Simple unigram and bigram methods were were used as the baseline for the restoration task. 10-fold cross-validation was applied in the evaluation of each of the models.

6 Results and Discussion

Our results (Table 6) indicate that with respect to the *n*-gram models, the embedding based diacritic restoration techniques perform comparatively well. Though the projected models (**IgboGNews** and **IgboEnBbl**) appear to have struggled a bit compared to the **IgboBible**, one can infer that having been trained originally with the same dataset and language of the task may have given the latter some advantage. It also captures all the necessary linguistic information for Igbo better than the projected models.

Again, **IgboEnBbl** did better than **IgboGNews** possibly because it was trained on a corpus that directly aligns with the Igbo data used in the experiment. The pre-trained **IgboWiki** model was abysmally poor possibly because, out of the 3111 entries in its vocabulary, 1,930 (62.04%) were English words while only 345 (11.09%) were found in our Igbo dictionary[10] used. It is not clear yet why all the results are the same across the methods. The best restoration technique across the models is the *Tweak3* which suggests that very frequent common words may have introduced some noise in the training process.

[9] The average number of instances is 1341 with the minimum and maximum numbers being 38 and 14,157 respectively.

[10] We note however that our Igbo dictionary was built from only the Igbo bible data and therefore is by no means complete. Igbo words and misspellings in **IgboWiki** that are not found in **IgboBible** vocabulary were simply dropped

Baselines: *n*-gram models

	Unigram	*Bigram*
	72.25%	80.84%

Embedding models

	Trained		*Projected*	
	IgboBible	**IgboWiki**	**IgboGNews**	**IgboEnBbl**
Basic	69.28	18.94	57.57	64.72
Tweak1	74.11	18.94	61.10	69.88
Tweak2	78.75	18.94	67.28	74.84
Tweak3	**82.49**	18.94	**72.98**	**76.34**

Table 6: Accuracy Scores for the Baselines, *Trained* and *Projected* embedding models [Bolds indicate best tweaking method].

7 Conclusion and Future Research Direction

This work contributes to the IgboNLP[11] (Onyenwe et al., 2018) project with the ultimately goal to build a framework that can adapt, in an effective and efficient way, existing NLP tools to support the development of Igbo. This paper addresses the issue of building and projecting embedding models for Igbo as well as applying the models to diacritic restoration.

We have shown that word embeddings can be used to restore diacritics. However, there is still room for further exploration of the techniques presented here. For instance, we can investigate how generalizable the models produced are with regards to other tasks e.g. sense disambiguation, word similarity and analogy tasks. On the restoration task, the design here appear to be more simplistic than in real life as one may want to restore an entire sentence, and by extension a document, and not just fill a blank. Also, with Igbo being a morphologically rich language, the impact of character and sub-word embeddings as compared to word embeddings could be investigated.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

John Cocks and Te-Taka Keegan. 2011. A Word-based Approach for Diacritic Restoration in Māori. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 126–130, Canberra, Australia. Url = http://www.aclweb.org/anthology/U/U11/U11-2016.

David Crandall. 2005. Automatic Accent Restoration in Spanish text. [Online; accessed 7-January-2016].

Guy De Pauw, Gilles-Maurice De Schryver, L. Pretorius, and L. Levin. 2011. Introduction to the Special Issue on African Language Technology. *Language Resources and Evaluation*, 45:263–269.

Guy De Pauw, Peter W Wagacha, and Gilles-Maurice De Schryver. 2007. Automatic Diacritic Restoration for Resource-Scarce Language. In *International Conference on Text, Speech and Dialogue*, pages 170–179. Springer.

Ignatius Ezeani, Mark Hepple, and Ikechukwu Onyenwe. 2016. Automatic Restoration of Diacritics for Igbo Language. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016, Brno , Czech Republic, September 12-16, 2016, Proceedings*, pages 198–205, Cham. Springer International Publishing.

Ignatius Ezeani, Mark Hepple, and Ikechukwu Onyenwe. 2017. Lexical Disambiguation of Igbo using Diacritic Restoration. *SENSE 2017*, page 53.

Jerid Francom and Mans Hulden. 2013. Diacritic error detection and restoration via part-of-speech tags. *Proceedings of the 6th Language and Technology Conference*.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. 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)*, volume 1, pages 1234–1244.

Rada Mihalcea. 2002. Diacritics restoration: Learning from letters versus learning from words. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 339–348, London, UK, UK. Springer-Verlag.

[11] See igbonlp.org

Ikechukwu E Onyenwe, Mark Hepple, Uchechukwu Chinedu, and Ignatius Ezeani. 2018. A Basic Language Resource Kit Implementation for the Igbonlp Project. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 17(2):10:1–10:23.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Kevin P. Scannell. 2011. Statistical unicodification of african languages. *Language Resource Evaluation*, 45(3):375–386.

Michel Simard. 1998. Automatic Insertion of Accents in French texts. *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*, pages 27–35.

Udemmadu Thecla-Obiora. 2012. The issue of ambiguity in the igbo language. *AFRREV LALIGENS: An International Journal of Language, Literature and Gender Studies*, 1(1):109–123.

Dan Tufiş and Adrian Chiţu. 1999. Automatic Diacritics Insertion in Romanian Texts. *Proceedings of the International Conference on Computational Lexicography*, pages 185–194.

Nikola Šantić, Jan Šnajder, and Bojana Dalbelo Bašić. 2009. Automatic Diacritics Restoration in Croatian Texts. In *The Future of Information Sciences, Digital Resources and Knowledge Sharing*, pages 126–130. Dept of Info Sci, Faculty of Humanities and Social Sciences, University of Zagreb , 2009. ISBN: 978-953-175-355-5.

Peter W. Wagacha, Guy De Pauw, and Pauline W. Githinji. 2006. A Grapheme-based Approach to Accent Restoration in Gĩkũyũ. In *In Proceedings of the fifth international conference on language resources and evaluation*.

David Yarowsky. 1994. A Comparison of Corpus-based Techniques for Restoring Accents in Spanish and French Text. In *Proceedings, 2nd Annual Workshop on Very Large Corpora*, pages 19–32, Kyoto.

David Yarowsky. 1999. Corpus-based techniques for restoring accents in spanish and french text. In *Natural Language Processing Using Very Large Corpora*, pages 99–120. Kluwer Academic Publishers.