Automatic Restoration of Diacritics for Igbo Language

Ignatius Ezeani, Mark Hepple, and Ikechukwu Onyenwe

NLP Group, Department of Computer Science, The University of Sheffield, United Kingdom.

Abstract. Igbo is a low-resource African language with orthographic and tonal diacritics, which capture distinctions between words that are important for both meaning and pronunciation, and hence of potential value for a range of language processing tasks. Such diacritics, however, are often largely absent from the electronic texts we might want to process, or assemble into corpora, and so the need arises for effective methods for automatic diacritic restoration for Igbo. In this paper, we experiment using an Igbo bible corpus, which is extensively marked for vowel distinctions, and partially for tonal distinctions, and attempt the task of reinstating these diacritics when they have been deleted. We investigate a number of word-level diacritic restoration methods, based on n-grams, under a closed-world assumption, achieving an accuracy of 98.83% with our most effective method.

Key words: diacritic restoration, sense disambiguation, low resourced languages, Igbo language

1 Introduction

Diacritics are simply defined as marks placed over, under, or through a letter in some languages to indicate a different sound value from the same letter without the diacritics[1]. The word "diacritics" was derived from the Greek word *diakritikós*, meaning "distinguishing".

Although English does not have diacritics (apart from some few borrowed words), many of the worlds language groups (Germanic, Celtic, Romance, Slavic, Baltic, Finno-Ugric, Turkic etc), as well as many African languages, use a wide range of diacritized letters in their orthography.

Automatic Diacritic Restoration Systems (ADRS) are tools that enable the restoration of missing diacritics in texts. Many forms of such tools have been proposed, designed and developed. Some ADRS restore diacritics on existing texts while others insert appropriate diacritics "on-the-fly" during text creation [9] but not much has been done for Igbo language.

[1] http://www.merriam-webster.com/dictionary/diacritic

1.1 Igbo Writing System and Diacritics

Igbo, one of the three major Nigerian languages and the primary native language of the Igbo people of southeastern Nigeria, is spoken by over 30 million people mostly resident in Nigeria and are of predominantly Igbo descent. It is written with the Latin scripts and has many dialects. Most written works, however, use the official orthography produced by the *Ọnwụ Committee*[2].

The orthography has 8 vowels (*a, i, o, u, ị, ọ, ụ*) and 28 consonants (*b, gb, ch, d, f, g, gw, gh, h, j, k, kw, kp, l, m, n, nw, ny, ṅ, p, r, s, sh, t, v, w, y, z*). Some researchers, however, consider the *Ọnwụ* orthography inadequate because of the inability to represent many dialectal sounds with it [1].

In Table 1, the Igbo letters with orthographic or tonal (or both) diacritics are presented with their diacritic forms and some examples of how they can change the meanings of the words they appear in[3].

Char	Ortho	Tonal	Examples
a	–	à,á, ā	*ákwà*(cloth), *àkwà*(bed/bridge), *ákwá*(cry),*àkwá*(egg)
e	–	è,é, ē	*égbè*(gun), *égbé*(kite),
i	ị	ì, í, ī, ị̀, ị́, ị̄	*ísí*(head), *ísì*(smell), *ísī*(to cook), *ịsị̄*(to say)
o	ọ	ò, ó, ō, ọ̀, ọ́, ọ̄	*ólù*(neck), *ọ́lụ̀*(work); *ódō*(pestle), *ọ̀dọ́*(pool)
u	ụ	ù, ú, ū, ụ̀, ụ́, ụ̄	*égwú*(dance/song), égwù(fear)
m	–	m̀,ḿ, m̄,	*ḿmádù*(a person), *m̀bèrèdé* (accident)
n	ṅ	ǹ,ń, n̄,	*ńdụ̀*(life), *ǹdò* (shelter)

Table 1. Diacritics in Igbo language

2 Problem Definition

Lack of diacritics can often lead to some semantic ambiguity in written Igbo sentences. Although a human reader can, in most cases, infer the intended meaning from context, the machine may not. Consider the following statements and their literal translations:

Missing orthographic diacritics

1. *Nwanyi ahu banyere n'**ugbo** ya.* (The woman entered her [**farm|boat/craft**])
2. *O kwuru banyere **olu** ya.* (He/she talked about his/her [**neck/voice|work/job**])

[2] http://www.columbia.edu/itc/mealac/pritchett/00fwp/igbo/txt_onwu_1961.pdf

[3] Observe that *m* and *n*, nasal consonants, are sometimes treated as tone marked vowels.

Missing tonal diacritics

1. *Nwoke ahu nwere* ***egbe*** *n'ulo ya.* (That man has a [**gun|kite**] in his house)
2. *O dina n'elu* ***akwa****.* (He/she is lying on the [**cloth|bed,bridge|egg|cry**])
3. ***Egwu*** *ji ya aka.* (He/she is held/gripped by [**fear|song/dance/music**])

As seen above, ambiguities may arise when diacritics – orthographic or tonal – are omitted in Igbo texts. In the first examples, **ugbo**(*farm*) and **ụgbọ**(*boat/craft*) as well as **olu**(*neck/voice*)and **ọlụ**(*work/job*) were candidates in their sentences.

Also, in the second examples, **égbé**(*kite*) and **égbè**(gun); **ákwà**(cloth), **àkwà** (bed or bridge), **àkwá**(egg) (or even **ákwá**(cry) in a philosophical or artistic sense); as well as **égwù**(fear) and **égwú**(music) are all qualified to replace the ambiguous word in their respective sentences.

The examples above incidentally showed words that belong to the same class i.e. nouns. However, instances abound of keywords (i.e. non-diacritic variant of a word) that represent actual forms that span different classes. For example, in the first two sentences, **banyere** could mean *bànyèrè* (*entered*, a verb) or *bànyéré* (*about*, a preposition). A high-level description of the proposed system is presented in Fig 1 below.

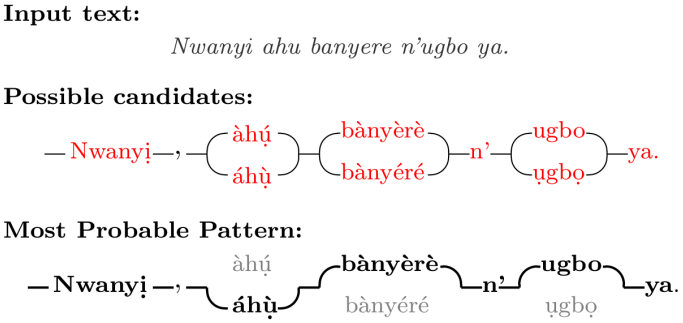

Fig. 1. Illustrative View of the Diacritic Restoration Process

3 Related Literature

Two common approaches to diacritic restoration are highlighted in this paper: *word based* and *character based*.

3.1 Word level diacritic restoration

Different implementation schemes of the word-based approach have been described e.g. Simard [9] adopted successive operations of *segmentation*, *hypothesis*

generation and *disambiguation* using POS-tags and HMM language models for French texts. The work was later extended to an accent insertion tool capable of "on-the-fly-accentuation" (OTFA), on text editors. On the Croatian language, Šantić *et al* [7] applied the use of substitution schemes, a dictionary and language models in implementing a similar architecture. It involves *tokenisation*, *candidate generation*, and *correct form selection*.

Yarowsky [12] also used dictionaries with decision lists, Bayesian classification and Viterbi decoding based on the surrounding context. Crandall [3], using Bayesian approach, HMM and a hybrid of both, as well as different evaluation method, attempted to improve on Yarowsky's work on Spanish text and reported a restoration accuracy of 7.6 errors per 1000 words. Cocks and Keegan [2], with a multi-genre corpus of about 4.2 million Maori words used naive Bayes classifiers and extracted word-based n-grams relating to the target word as instance features. Tufiş and Chiţu [10] also applied POS tagging to restore diacritics in Romanian texts and subsequently dealt with "unknown words" using character based back-off.

3.2 Grapheme or letter level diacritic restoration

Mihalcea *et al.* [5], with their work on Romanian, presented an argument for letter based diacritic restoration for low resource languages. With a 3 million word corpus from the online Romanian newspaper, they implemented an instance based and a decision tree classifiers which gave a letter-level accuracy of 99% implying a much lower accuracy at the word-level.

This approach became popular among developers of language tools for low resourced languages. However, Wagacha *et al* [11], replaced the evaluation method in Mihalcea's work, with a more appropriate word-based approach while De Pauw [4] used, the "lexical diffusion" (*LexDif*)[4] metric to quantify the disambiguation challenges on language bases. Each of these works recorded an accuracy level over 98%.

3.3 Igbo Diacritic Restoration

The only attempt we know of at restoring Igbo diacritics is reported by Scannell [8] in which a combination of word- and character-level models were applied. For the word-level, they used two lexicon lookup methods, *LL* which replaces ambiguous words with the most frequent word and *LL2* that uses a bigram model to determine the output. They reported accuracies of 88.6% and 89.5% for Igbo language on the *LL* and *LL2* models respectively.

[4] *LexDif* is the average number of candidates per wordkey, calculated by dividing the total word types with the unique wordkeys. A wordkey is gotten by stripping the diacritics off a word.

4 Experimental Setup

We present the preliminary report on the implementation of some word-level Igbo ADRS models based on ideas from the work of Scannell [8]. We developed two broad categories of n-gram - bigrams and trigrams - models with their variants.

4.1 Experimental Methods

In this work, we aimed at extending the work of Scannell [8] on the word-level diacritic restoration of Igbo language. Some key distinctions of the approach we used from their works are highlighted below:

Data size: Their experiment used a data size of 31k tokens with 4.3k word types while ours used 1.07m with 14.4k unique tokens. Table 2 shows percentage distributions of some ambiguous words in the text.

Preprocessing: Our tokenizer considered certain language information[5] that their method may not have considered.

Baseline model: Their work reported a baseline measure on the stripped version which is our *bottom line* measure. Our baseline model is similar to their first lexicon lookup *LL* model.

Bigram models: Our work extended their best model (bigram) with different smoothing techniques, *words vs keys* approach, and *backward replacement* features

Trigram models: Trigram models were not included in their work but we implemented different trigram models with similar structures as the bigram models.

4.2 Experimental Data

how do you get 1.0398???

The Igbo Bible corpus[6] used for this work contains 1,070,429 tokens (including punctuations and numbers) with 14,422 unique word types out of which 5580 are unambiguous (i.e. appeared only in one diacritic form) in the text. The lexical diffusion on the text is 1.0398. A bible verse (not a sentence) was used as the unit of data entry.

Bigram and trigram counts were generated while a non-diacritic version of the corpus was created by stripping off all diacritics from the main bible text. An outright evaluation of the stripped text yields a bottom line accuracy of 71.30%.

Our task involves creating the non-diacritic version, generating a look-up dictionary of unique key entries and their diacritic variants based on a closed-world[7] assumption, applying a restoration model to the stripped version one line

[5] For example: strings like *"na"*, (mostly conjunction), *"na-"* (auxiliary) or *"n' "* (preposition) are treated as valid tokens due to the special roles the symbols play in distinguishing the word classes.

[6] This corpus was originally processed by Onyenwe *et al.*[6]

[7] Since we did not deal with unknown words, we simplified our models by assuming that words not found in our dictionary do not exist.

Key	*Words*	*# of occurrences*	*%age*
na	**na**	28041	95.41%
	ná	1349	4.59%
o	**o**	7477	24.75%
	ó	8	0.03%
	ọ́	252	0.83%
	ò	83	0.27%
	ọ̀	1053	3.49%
	ọ	21339	70.63%
ruru	**ruru**	225	49.34%
	rụrụ	231	50.66%
agbago	**agbago**	49	51.58%
	agbagọ	46	48.42%
bu	**bu**:	241	1.49%
	bú:	2	0.01%
	bụ́:	6050	37.39%
	bụ:	9887	61.11%

Table 2. Sample percentage distribution of some of the ambiguous words

at a time and keeping a count of the correctly restored tokens. The performance evaluation is measured by computing the percentage of the entire tokens that are correctly restored.

4.3 Model descriptions

Baseline models: *a01* and *a02* As stated earlier, the bottom line model (a01) compared every diacritic token with its corresponding non-diacritic key from the stripped text. The baseline (a02) applied a simple unigram model that picks the most occurring candidate from the data.

Bigram models: *b01* ... *b06* Given a stripped word (or wordkey), these models use maximum likelihood estimation (MLE) to select the most probable word given the previous word. *b01* and *b02* are the "one key" variants. They use only the wordkey of the current word to generate candidates while retaining the previously restored preceding word. They differ in the smoothing techniques (Add 1 and Backoff). *b03* to *b06"* are the "two key" variants i.e. for every bigram, the two wordkeys will be used to find all possible candidates of both words and then select the most probable combination. This is motivated by the "assumption" that an error may have occurred in the previous step and should not be carried along. Also a technique called *backward replacement* was introduced to provide an opportunity to "step back and correct"[8] any assumed error as we go along i.e. if the most probable bigram suggest a different diacritic form for the preceding word, then it will be replaced with the current word.

[8] We recognise that this might be counter productive as correctly restored words in the previous step may be wrongly replaced again in the next.

Trigram models: *t07 ... t12* These are the trigram versions of the models described above. The "one key" variants, *t07* and *t08* use the last two restored words and the candidates of the given key to get the most probable trigram from the data while the rest generate fresh candidates with the current and the two preceding keys. The smoothing techniques as well as the backward replacement methods were also tested on these models.

Models	Accuracy	Amb Acc	%Impr
a01:Bottomline: Non-diac Text	71.30%	–	–
a02:Baseline: Most frequent	96.23%	91.52%	86.86%
b01:Bigram-1Key+Add1:	97.21%	94.45%	90.28%
b02:Bigram-1Key+Backoff:	97.75%	94.18%	92.16%
b03:Bigram-2Key+Add 1:	97.36%	94.60%	90.80%
b04:Bigram-2Key+Add 1-BR:	97.48%	94.85%	91.22%
b05:Bigram-2Key+Backoff:	97.77%	94.36%	92.23%
b06:Bigram-2Key+Backoff-BR:	96.17%	90.14%	86.66%
t07:Trigram-1Key+Add 1:	97.66%	92.94%	91.85%
t08:Trigram-1Key+Backoff:	92.01%	77.96%	72.16%
t09:Trigram-3Key+Add 1:	98.46%	95.91%	94.63%
t10:Trigram-3Key+Backoff:	92.86%	81.37%	75.12%
t11:Trigram-3Key+Add 1-BR:	**98.83%**	**97.57%**	**95.92%**
t12:Trigram-3Key+Backoff-BR:	92.29%	77.99%	73.14%

Table 3. Results of experiments using the different models

4.4 Results, Conclusion and Future Work

This paper describes a knowledge-light language independent method for diacritic restoration for Igbo texts using n-gram models with a comparison of smoothing and replacement techniques under a closed-world assumption. The baseline method used a unigram model with an accuracy of 96.23%. The results show that the trigram models generally outperform the bigram models.

While the Add 1 smoothing technique improves as the experiment progressed, the Backoff seems inconsistent, beating the Add 1 with the bigrams and dropping in performance with the trigrams. Backward replacement is introduced and it seems to work though it is not yet clear why it does. However, while it has boosted the performance of the Add 1 at each stage, it has clearly deteriorated that of the Backoff model.

The "three key" trigram model with the *Add 1* and backward replacement is the most effective method with a performance accuracy of 98.83%. They outperformed the best models in literature but future works will attempt to improve on the robustness with an open-world assumption while exploring the backward replacement and other smoothing techniques. Expanding the data size across multiple genre and handling "unknown words" will form the main focus of the

next experiments. We also intend to investigate the effects of POS-tagging and a morphological analysis on the performance of the models and explore the connections between this work and the broader field of word sense disambiguation.

Acknowledgments. Many thanks to Nnamdi Azikiwe University & TETFund Nigeria for the funding, my colleagues at the IgboNLP Project, University of Sheffield, UK and Prof. Kelvin P. Scannell, St Louis University, USA.

References

1. Achebe, I., Ikekeonwu, C., Eme, C., Emenanjo, N. and Wanjiku, N.:A Composite Synchronic Alphabet of Igbo Dialects (CSAID), IADP, New York, 2011.
2. Cocks J., Keegan T.: A Word-based Approach for Diacritic Restoration in Māori, Proceedings of the Australasian Language Technology Association Workshop 2011, December, 2011, Canberra, Australia, 126–130.
3. Crandall, D., Automatic Accent Restoration in Spanish text, 2016 http://www.cs.indiana.edu/ djcran/projects/674_final.pdf, "[Online; accessed 7-January-2016]"
4. De Pauw, G., De Schryver, G. M., Pretorius, L., Levin L.: Introduction to the Special Issue on African Language Technology, Language Resources and Evaluation, vol. 45, 263-269, Springer Online, 2011.
5. Mihalcea, R.F.: Diacritics restoration: Learning from letters versus learning from words, In: Gelbukh, A. (ed.) CICLing LNCS, 2002, Springer, Heidelberg, 2276, 339–348
6. Onyenwe, I. E., Uchechukwu C., Hepple M. R.: 2014 Part-of-speech Tagset and Corpus Development for Igbo, an African Language LAW VIII - The 8th Linguistic Annotation Workshop. ACL, Dublin, Ireland, 2014, 93–98
7. Šantić, N., Šnajder, J., Dalbelo Bašić, B.: Automatic Diacritics Restoration in Croatian Texts, The Future of Information Sciences, Digital Resources and Knowledge Sharing, Stančić, Hrvoje and Seljan, Sanja and Bawden, David and Lasić-Lazić, Jadranka and Slavić, Aida, 2009, 126-130, ISBN: 978-953-175-355-5
8. Scannell, K. P.: Statistical Unicodification of African Languages, Language Resource Evaluation, 45, 3, Sep 2011, 375–386, Springer-Verlag New York, Inc.,Secaucus, NJ, USA,
9. Simard, M.: Automatic Insertion of Accents in French Texts, Proceedings of the Third Conference on Empirical Methods in Natural, Language Processing, 1998, 27–35
10. Tufiş, D., Chiţu, A.: Automatic Diacritics Insertion in Romanian Texts, Proc Int Conf on Comput Lexicography, Pecs, Hungary, 185–194, 1999
11. Wagacha P. W., De Pauw G., Githinji, P. W.: A Grapheme-based Approach to Accent Restoration in Gĩkũyũ, In Proc 5th Int. Conf on language resources and evaluation, 2006
12. Yarowsky, D.: Corpus-based Techniques for Restoring Accents in Spanish and French Text. Nat Lang Processing Using Very Large Corpora, Kluwer Academic Publishers 99-120, (1999)