**Background**

**Basaa**

Basaa is a Bantu language spoken in Cameroon that uses high and low tones contrastively (Hyman, 2003). Official orthographies were established and published in a bilingual guide for all Cameroonian languages in the 1970s (Tadadjeu and Sadembouo, 1984). The purpose of this guide was to unify and harmonize the alphabets of all languages within Cameroon, and to expand the available literature in and to promote the teachings of Cameroonian languages. Within Tadadjeu and Sadembouo’s 1984 guide, there is a section dedicated to tone marking. Although a system was already in place, it was not systematic across languages. Tadadjeu and Sadembouo’s guide designated six tones: the high tone (á), the mid tone (ā ), the low tone (à), the rising tone (ǎ), the falling tone (â), and a central low tone, which is not used in Basaa (1984). These tones are marked because they are contrastive. That is, the meaning of the word can change just by altering a single tone. Furthermore, Basaa does not have a fixed tone system, and processes like High Tone Spreading make the need for systematic tone marking even more crucial (Hyman, 2003). Currently texts in Basaa are limited, but those that are produced according to the standards set out by the Academy of Languages of Cameroon differ in their orthography from those texts created by missionary groups, specifically in their tonal markings, making the creation of a larger homogenous corpus of Basaa language for for experimental or pedagogical purposes impossible without reconciling these differences.

**Problem definition**

For some languages, like Igbo, human readers can generally understand its written form without the use of diacritics (though see Ezeani, Hepple, and Onyenwe, 2017 for examples of ambiguity), but machines cannot reliably read the same text without diacritics. This can lead to impoverished technological resources for speakers of these languages because diacritic restoration is important for improving speech recognition software, machine translation systems (e.g. Google translate), and text generation (Ezeani, Hepple, & Onyenwe, 2017). While work has been done to improve diacritic restoration for Igbo (De Pauw, Wagacha, & De Schryver, 2007**,** Ezeani, Hepple, & Onyenwe, 2016, 2017), to our knowledge, no work has been done to date on Basaa.

In order to create and  implement a lexical disambiguation process like the ones mentioned above, **systems** usually rely on a large, ever expanding corpus of data. However corpuses for languages like Basaa are sparse. Furthermore, within the few resources that are available, there is a divergence of diacritic use. So, the lexical disambiguation process for Basaa and other languages with similar situations must utilize an approach that can both quickly and accurately detect and resolve these discrepancies without relying on a vast homogenous corpus for training.

**Approaches**

**Rule-based approach**

The used methods include cascading Weighted Finite-State Transducers[33], lexicon retrieval and rule-based morphological analysis [7]. One other particular work [9] used diacritized text borrowing from other sources to diacritize a highly cited text. [Article 1] To apply this, we have to identify rules (they probably exist, but still).

Rule-based corrections are linked to the input and output of the model to apply some changes to the output. These rules can select the appropriate diacritic for some characters in some contexts, or exclude the wrong choices in other contexts by nullifying their probabilities. Different sets of rules are applied to the outputs to eliminate some impossible diacritizations according to Arabic rules [Article 1]

Some researchers have developed systems to restore diacritics based on the morphological, syntactic, and semantic rules of Arabic [10], [11]. These rules are systematic, but complex [12], [13]. Morphological rules decompose the undiacritized word into its morphological entities: the class of the word, prefix, root, form, and suffix. Syntactic rules determine which diacritics accompany the last character of the word. Semantic rules resolve ambiguous cases. Rule-based diacritization is vocabulary independent; however, it is complex, time-consuming, and difficult to regularly update the rules or extend them to other Arabic dialects [14]. [article 5]

As there are few linguistic resources describing the structure of Basaa, this type of approach does not seem to be the most effective.

**Statistical approach**

Statistical approaches include using Hidden Markov models both on word level and on character level [8,18,21], N-grams models on word level and on character level as well [10], Dynamic Programming methods [24–26], classical Machine learning models such as Maximum-entropy classifier [46], and Deep Learning methods like the Deep Neural Networks, both the classical Multi-Layer Perceptron and the advanced Recurrent Neural Networks[6,14,32,36].

Markov models are restrictive in the diacritization process because they match each observation to a distinct state within the model, whereas, in reality, each state (diacritic) has a probabilistic function that generates the possible observations (ASCII codes of the script letters) [Article 5]

This program passes through three modules. The first module reads the text line and transfers it into a sequence of predefined observation symbols. The second module generates the output sequence: the diacritics. The final module arranges the output sequence with the raw text by imposing the diacritics above and below the equivalent characters. [Article 5]

[Article 1] Ngrams at both the word and character levels give an approach to vectorize a sentence and are promising approaches for our project on Basaa.

**Hybrid approach**

Hybrid approaches use a deep learning model which is a multi-layer recurrent neural network with LSTM and Dense layers, a character-level rule-based corrector which applies deterministic operations to prevent some errors, and a word-level statistical corrector which uses the context and the distance information to fix some diacritization issues. [Article 1]

The output and the input of the previous phase are transformed and merged to generate a standard diacritized sentence. The sentence is segmented into space delimited words and augmented by unique starting and ending entities. Every word in the sentence is checked up to 4 times in the levels of correction using the saved training data. If any acceptable correction is found at any level, the word will be corrected. Otherwise, it will be forwarded to the next level. In the case where many corrections get the same score in a single level, the first correction is chosen. If no correction is found, the predicted diacritization of the previous component is not changed. [Article 1]

Combining linguistic knowledge with stochastic techniques has resulted in a hybrid system that determines the most likely diacritics for a given undiacritized text, then factorizes each Arabic word into all its possible morphological constituents, and finally shortlists the sequence of morphemes that is the most likely diacritization [1], [26]. [Article 5]

**Level of application**

These three approaches (rule-based, statistical, and hybrid) can be used at the word-level, morpheme-level, or grapheme-level.

**Word-level**

Well-resourced languages, like Serbian or Arabic, tend to rely on word-based methods for diacritic restoration, but these are language specific and rely on a large corpus and NLP tools (Krstev, Stankovic, & Vitas, 2018). For example, Abbad and Xiong (2020) approach Arabic diacritic restoration first using trigrams, then bigrams and, finally, a Minimum Edit Distance Correction]

There are rule based [4], [7] as well as statistical methods for diacritics restoration. Statistical methods based on Hidden Markov Models (HMMs) were introduced in [8]. The hidden states are the diacritized word and the observations are the undiacritized words. The transition and emission probabilities can be estimated from a large training corpus [9]. The most likely diacritized word sequence can be found using the Viterbi [Article 3]

Preliminary results on a public domain corpus show that dynamic programming decoding based on higher order n-gram models can lead to better results than bigram models. [Article 4]

Like Basaa, Igbo is a low resource tonal language in need of diacritic restoration. Typically, grapheme-level approaches are the most successful for low resource languages because there are not many NLP tools or large corpora available. However, Ezeani, Hepple, and Onyenwe (2016, 2017) developed and tested word-level approaches to restore diacritics for Igbo text. More specifically, a decision tree machine-learning algorithm was found to be the most effective (average accuracy of 88.64%) followed by a K Nearest Neighbors algorithm (Ezeani et al., 2017). This approach seems like an avenue to explore more for our project.

**Morpheme-level**

Arabic is a morphologically rich language and has a high OOV rate [28]. One way to solve this problem is to use character level [29] or n-gram character level [30] diacritization systems. In this letter, we use a novel approach to support open vocabulary diacritics restoration based on the Byte Pair Encoding (BPE) method [31], [32]. The BPE method segments the words into variable length sub-word units and allows open vocabulary from fixed sub-word units dictionary. We test this approach within the language model scoring framework and report improved performance. [Article 3]

The BPE method segments the words into variable length sub-word units and allows open vocabulary from a fixed sub-word units dictionary. The results on the Tashkeela diacritization task show that the proposed method outperforms the character based methods commonly used to support open vocabulary diacritics restoration within the language model scoring framework. [Article 3]

**Grapheme-level**

The input is mapped to a set of 38 numeric labels representing all the Arabic characters in addition to 0 and the white space. It is transformed into a 2D one-hot encoded array, where the size of the first dimension equals the length of the sentence, and the size of the second equals the number of the labels. After that, this array is extended to 3 dimensions by inserting the time steps dimension as the second dimension and moving the dimension of the label into the third position. The time steps are generated by a sliding window of size 1 on the first dimension. The number of time steps is fixed to 10 because this number is large enough to cover most of the Arabic words, along with a part of their previous words. The output of the primary diacritics is also transformed from a vector of labels to a 2D one-hot array. [Article 1] Might be a way to vectorize each sentence.

**An approach to try [Article 3]**

A large diacritized training corpus is segmented into tokens. This tokenized corpus is used to estimate the n-gram language model. The unique diacritized tokens in the training set is extracted to construct the mapping dictionary. This dictionary (keys are the undiacritized tokens and values are the diacritized tokens) is used during the test phase to generate the possible diacritized tokens from the input undiacritized token sequence. For an input undiacritized token sequence, generate all the possible diacritized hypotheses using the unique tokens dictionary. Score all the hypotheses and select the hypothesis (diacritized tokens) with the highest score. Map the tokens into the actual diacritized words

**References**

Abbad, H., & Xiong, S. (2020). Multi-components System for Automatic Arabic Diacritization.

*Advances in Information Retrieval*, 12035, 341-355.

Adadjeu, M. & Sadembouo, E. (1979). Alphabet générale des langues Camerounaises. Departement des Langues Africaines et Linguistique, Université de Yaoundé, Cameroun.

Adali, K., & Eryiğit, G. (2014). Vowel and diacritic restoration for social media texts.

*Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM),* pp. 53-61.

Asahiah, F. O. L., BÍ, O., Tunjiajadi, O., & Adagunodo, E. R. (1999). A Survey of Approaches

to Diacritic Restoration. *Natural Language Engineering*, *1*(1), 1-23.

Chennoufi, A., & Mazroui, A. (2017). Morphological, syntactic and diacritics rules for automatic

diacritization of Arabic sentences. *Journal of King Saud University-Computer and Information Sciences*, *29*(2), 156-163.

De Pauw, G., Wagacha, P. W., & De Schryver, G. M. (2007). Automatic diacritic restoration for

resource-scarce languages. *International Conference on Text, Speech and Dialogue*. Springer, Berlin, Heidelberg, pp. 170-179

Ezeani, I., Hepple, M., & Onyenwe, I. (2016). Automatic restoration of diacritics for Igbo

language. *International Conference on Text, Speech, and Dialogue.* Springer, Cham, pp. 198-205.

Ezeani, I., Hepple, M. R., & Onyenwe, I. (2017). Lexical disambiguation of Igbo using diacritic

restoration. *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications,* pp. 53-60.

Francom, J., & Hulden, M. (2013). Diacritic error detection and restoration via part-of-speech

tags. *Proceedings of the 6th Language and Technology Conference*.

<https://www.researchgate.net/profile/Jerid_Francom/publication/263028685_Diacritic_error_detection_and_restoration_via_part-of-speech_tags/links/0a85e53a19723eb241000000.pdf>

 Luu, T. A., & Yamamoto, K. (2012, November). A pointwise approach for Vietnamese

diacritics restoration. 2012 International Conference on Asian Language Processing, pp. 189-192.

Hifny, Y. (2012). Higher order n-gram language models for Arabic diacritics restoration. In *The*

*Twelfth Conference on Language Engineering*, pp. 6-12.

Hifny, Y. (2019). Open Vocabulary Arabic Diacritics Restoration. *IEEE Signal Processing*

*Letters*, 26, 1421-1425.

Hyman, L. M. (2003). Basaá (A43). *The Bantu languages*, *257*, 282.

Khorsheed, M. S. (2018). Diacritizing Arabic text using a single hidden Markov model. *IEEE*

*Access*, *6*, 36522-36529.

Krstev, C., Stankovic, R., & Vitas, D. (2018, May). Knowledge and rule-based diacritic

restoration in Serbian. *Proceedings of the Third International Conference Computational Linguistics in Bulgaria*, pp. 41-51.

Masmoudi, A., Mdhaffar, S., Sellami, R., & Belguith, L. H. (2019). Automatic diacritics

restoration for Tunisian dialect. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, *18*(3), 1-18.

Novák, A., & Siklósi, B. (2015). Automatic diacritics restoration for Hungarian.

*Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2286-2291.

Torunoğlu-Selamet, D., & Eryiğit, G. (2014). A cascaded approach for social media text

normalization of Turkish. *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pp. 62-70.

Zayyan, A. A. M. (2017). *An automatic diacritization algorithm for undiacritized Arabic text*

[Unpublished doctoral dissertation]. Universiti Utara Malaysia.

Zitouni, I., & Sarikaya, R. (2009). Arabic diacritic restoration approach based on maximum

entropy models. *Computer Speech & Language*, 23(3), 257-276.