applied sciences

MDPI

Article

Better Word Representation Vectors Using Syllabic Alphabet: A Case Study of Swahili

Casper S. Shikali [1,2], Zhou Sijie [1,*], Liu Qihe [1] and Refuoe Mokhosi [1]

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Xiyuan Ave, West Hi-Tech Zone, Chengdu 611731, China; cshikali2002@yahoo.com (C.S.S.); qiheliu@uestc.edu.cn (L.Q.); refuoemokhosi@yahoo.com (R.M.)

[2] School of Information and Communication Technology, South Eastern Kenya University, Kitui 170-90200, Kenya

* Correspondence: sjzhou@uestc.edu.cn

Received: 22 July 2019; Accepted: 29 August 2019; Published: 4 September 2019

Featured Application: This work is applicable in computer science, software engineering and computational linguistic specifically in natural language processing.

Abstract: Deep learning has extensively been used in natural language processing with sub-word representation vectors playing a critical role. However, this cannot be said of Swahili, which is a low resource and widely spoken language in East and Central Africa. This study proposed novel word embeddings from syllable embeddings (WEFSE) for Swahili to address the concern of word representation for agglutinative and syllabic-based languages. Inspired by the learning methodology of Swahili in beginner classes, we encoded respective syllables instead of characters, character n-grams or morphemes of words and generated quality word embeddings using a convolutional neural network. The quality of WEFSE was demonstrated by the state-of-art results in the syllable-aware language model on both the small dataset (31.229 perplexity value) and the medium dataset (45.859 perplexity value), outperforming character-aware language models. We further evaluated the word embeddings using word analogy task. To the best of our knowledge, syllabic alphabets have not been used to compose the word representation vectors. Therefore, the main contributions of the study are a syllabic alphabet, WEFSE, a syllabic-aware language model and a word analogy dataset for Swahili.

Keywords: syllabic alphabet; word representation vectors; deep learning; syllable-aware language model; perplexity; word analogy

1. Introduction

Natural language processing (NLP) relies on word embeddings as input for machine learning or deep learning algorithms. For decades, NLP solutions were restricted to machine learning approaches that trained on handcrafted, high dimensional and sparse features [1]. Nowadays, the trend is neural networks [2], which use dense vector representations. Hence, the superior results on NLP tasks is attributed to word embeddings [3,4] and deep learning methods [5]. Therefore, as observed by the authors of [6–9], improved performance of downstream NLP tasks is achieved by learning vector representation of words in language models. Quality word vectors are expected to capture syntactic and semantic similarities among words by addressing the similarities in surface form of words and the context [8]. This has motivated the transition from the conventional one-hot word representation to word representation [10] based on words and sub-word information (characters and morphemes). Despite Mikolov et al.'s [4] contribution of distributed word representation, the urge for even better word representation has led to composition of word embeddings from sub-word information such as characters [8,11,12], character n-gram [13] and morphemes [7,14,15].

However, with all these developments, deep learning is yet to be utilized in processing low resource languages [16,17], particularly the syllabic based languages, such as Swahili, Xhosa, Luhya, Shona, Kikuyu and Mijikenda. In fact, the East Africa integration initiative by the respective countries has encountered language barrier as a challenge in the quest for a common language [18], a problem that could be overcome by automated language systems. A probable solution is automating Swahili using deep learning models that can learn its syllabic alphabet to effectively represent the Swahili words, which are highly agglutinative [19]. To the best of our knowledge, no study has considered learning word representation from constituent syllables of words in syllabic-based languages. We are inspired by the Swahili language teaching methodology, which first introduces syllables, then two-syllable words, and lastly complex words and sentences [20]. For this reason, we propose syllabic based word embeddings (WEFSE) to match Swahili's complex word morphology, as opposed to using characters or morphemes. This study generated word embeddings from syllable embeddings (WEFSE) but differently from Assylbekov et al. [21], who used an external hyphenator to segment the words into syllables. We hypothesize that learning word representations from syllabic alphabets captures both semantic meaning of words and handles new words. We attribute this to the fact that the syllables in Swahili are the smallest meaningful semantic unit and are a subset of the morphemes [22]. Consider the following Swahili verbs:

1. anapojitakia (when he/she wants it for himself/herself);
2. aliandika (he wrote); and
3. atakikamata (he will catch it).

From these examples we note that Swahili is highly agglutinative, as it starts with a root word and creates a new word by adding syllables such as "a", "na", "po", "ji", "ki", "li", "ndi", "ka", "ta" and "ma". This indicates that the ultimate meaning of the verb *anapojitakia* is a culmination of the subject prefix ("a"), tense prefix ("na"), relative prefix ("po"), object prefix ("ji"), root ("taka") and extension ("ia"). The position of each syllable in the word also bears syntactic and semantic meaning. For example, the second syllable "na", "li" and "ta" in Examples 1, 2 and 3 connote present, past and future tenses, respectively. Intuitively, words with the same syllables have a similar context. The study's objective was to effectively represent Swahili words by capturing the compositional and contextual aspects.

There is a trend in NLP of optimizing performance of downstream tasks through high quality vectors for word representation [6,23]. Classical language models [23] use contextual word information from a large corpus to generate word embeddings, however these models were deficient in representing rare words and new words [24]. This led to compositional models, in particular, character-aware models [6,8,11–13,25,26] aimed at mitigating the data sparsity problem. Although these models address the rare word problem, characters carry no inherent semantic meaning. Alternatively, the compositional models based on morphemes [7,14,15,27–29] address the semantic meaning deficiency in character models. However, external fragmentation of words into morphemes propagate errors into the models, affecting the quality of the word embeddings [29]. Our work is similar to those of Assylbekov et al., Yu et al. and Mikolov et al. [9,21,30] on the basis of learning syllable and word representation. However, we utilized a defined syllabic alphabet instead of an external hyphenation algorithm to divide the words into syllables, which we hypothesize that may introduce errors.

The architecture of our model resembles that of Assylbekov et al. [21]. Both models apply a convolutional neural network [31] to extract features and compose the word embeddings, a highway network [32] to model interactions among the syllables and finally a recurrent neural network language model [3]. Our model is different in terms of how the words are encoded to syllables. It takes syllables as input and then extracts feature maps using a convolutional neural network to form the word embeddings. Then, the language model, which is made of a long short term memory [33], predicts the target words given the contextual words. The model has the potential to generalize unseen words and apply existing knowledge on new words because it is learning from a standard finite

syllabic alphabet, which is the basis of all Swahili words. We chose Swahili as the main language for the experiments because of its syllabic structure, agglutinative and polysemous features, and its popularity in East and Central Africa [17,19]. For comparison purposes, we performed experiments on Xhosa and Shona, which are syllabic but with limited scope. The quality of the generated word embeddings (WEFSE) is demonstrated by the perplexity values of the syllable-aware language model on both small and medium datasets developed by Gelas et al. [34]. The perplexity values are very competitive with the existing state-of-the-art models and outperform the character-aware counterparts. We further evaluated our word embeddings using the word analogy task to verify the quality of the word embeddings.

The main contributions of our study are as follows:

1. syllable alphabet;
2. word embeddings from syllable embeddings (WEFSE) (to the best of our knowledge, the first attempt to use syllabic alphabet);
3. syllable-aware language model; and
4. Swahili word analogy dataset.

The remaining sections of the paper are organized as follows. Section 2 discusses the Swahili language structure and introduces the syllabic alphabet. Section 3 reviews the previous works and Section 4 provides the details of the proposed model. We outline the experiments in Section 5 and discuss results and word analogy task in Section 6. We conclude in Section 7 and provide more details of the experiments in the Appendix A.

2. Swahili Language Structure

Swahili is one of the Bantu languages widely spoken in East and Central Africa, with two main dialects being Unguja (which is spoken in Zanzibar) and Mvita (which spoken in Mombasa) [35]. It is influenced by languages such as Arabic, Persian, German, Portuguese, English and French [35]. This explains the presence of a couple of loan words such as *shukrani* (thanks) and *polisi* (police). Swahili is also very contextual with high agglutinative and polysemous features [19,36]. The following two sentences demonstrate the polysemy feature of the word *panda*:

1. Walimuua ndege kwa panda (they killed a bird using a catapult); and
2. Walipanda mti (they climbed a tree or they planted a tree).

The Swahili morphology depends on prefixes and suffixes, which are syllables. It has a large noun class system with distinctive singular and plural forms [37], which is achieved by using syllable pairs, for example, *mtu* (person) and *watu* (people), or *kitabu* (book) and *vitabu* (books). In fact, Ng'ang'a [38] observed that synthetic and functional information can be derived from these connotation bearing affixes attached to nouns. However, verbs, pronouns, adjectives and demonstrations must match with the noun class to guarantee effective Swahili communication. In addition, Swahili verbs are very agglutinative, and may consist of subject, tense, relative and object prefixes in addition to roots and extensions. The following list provides a few examples of the verbal agglutinative components:

- subject prefix: a, m, ni, wa, tu, si, hatu;
- tense prefix: na, ta, li, me, ngeli, ja, singali;
- relative: po, ko, o, cho, mo, yo, vyo, lo;
- object prefix: wa, ni, ji, ku, i, ya, vi, zi;
- root: sema, kuja, ahidi, acha, aga, ajiri; and
- extension: ia, iwa, lia, shwa, ana, isha, kia.

Swahili Syllabic Alphabet

This section briefly introduces the finite Swahili syllabic alphabet, which forms the basis of all Swahili words. Swahili uses the English alphabet letters with the exception of x and q. The vowels a, e,

i, o, and u count as syllables and constitute the smallest syllabic unit [35]. Swahili syllables are derived from vowels and consonants of the alphabet, with a syllable normally consisting of a vowel preceded by one to three consonants. In special cases, the syllable is a single vowel or consonant. For example, in the words *mtu* (person) and *anakula* (he is eating), the starting letters "m" and "a" are special syllables. It is imperative that the position of the syllable in a word be preserved to maintain the syntactic and functional information. The following are the rules for Swahili syllabification [35,39]:

1. a consonant or a vowel preceded by a vowel is the start of a syllable;
2. a consonant (other than a semi-vowel) preceded by a consonant is the start of a syllable (for n, m and loan words);
3. all syllables end at the beginning of the next syllable or at the end of the word;
4. where a pre-consonantal nasal functions as a syllabic peak, a syllable is formed by a combination of two sounds;
5. a cluster of a consonant and a semi-vowel together with a vowel can also form a syllable; and
6. a segment of clusters of three with a vowel can form a syllable.

To derive the syllabic alphabet, we apply the above rules, appropriately combine the consonants with the vowels based on the list of letters provided by Masengo [40] and add the list of special syllables. It is important to note that new words or loan words can be generated from the alphabet. For example, before a proper Swahili word for television (*runinga*) was coined, it was and is still referred to as *televisheni* whose syllables can be found in the alphabet.

Table 1 outlines the Swahili syllabic alphabet.

Table 1. The Swahili syllabic alphabet.

mbwa	mbwe	mbwi	ndwa	ndwe	ndwi	ngwa	ngwe	ngwi	njwa	njwe	njwi	nywa
nywe	shwa	shwe	shwi	chwa	chwe	chwi	pwa	pwe	pwi	pwo	swa	swe
swi	twa	twe	twi	zwa	zwe	zwi	cha	che	chi	cho	chu	dha
dhe	dhi	dho	dhu	gha	ghe	ghi	gho	ghu	kha	khe	kho	khu
mba	mbe	mbi	mbo	mbu	nda	nde	ndi	ndo	ndu	nga	nge	ngi
ngo	ngu	ng'a	ng'e	ng'o	nja	nje	nji	njo	nju	nya	nye	nyi
nyo	nyu	sha	she	shi	sho	shu	tha	the	thi	tho	thu	vya
vye	vyo	bwa	bwe	bwi	gwa	gwe	gwi	jwa	jwe	jwi	kwa	kwe
kwi	lwa	lwe	lwi	mwa	mwe	mwi	nza	nze	nzi	nzo	nzu	ba
be	bi	bo	bu	da	de	di	do	du	fa	fe	fi	fo
fu	ga	ge	gi	go	gu	ha	he	hi	ho	hu	ja	je
ji	jo	ju	ka	ke	ki	ko	ku	la	le	li	lo	lu
ma	me	mi	mo	mu	na	ne	ni	no	nu	pa	pe	pi
po	pu	ra	re	ri	ro	ru	sa	se	si	so	su	ta
te	ti	to	tu	va	ve	vi	vo	vu	wa	we	wi	wo
wu	ya	ye	yi	yo	yu	vu	za	ze	zi	zo	zu	a
e	i	o	u	b	d	f	k	m	n	s	–	–

3. Related Work

In this section, we outline the previous works on Swahili and those that are in context with deep learning methods and word representation for NLP.

3.1. Swahili Natural Language Processing

Efforts have been made to include Swahili among the automated languages in NLP. Most of the works are characterized by morphological analysis to resolve ambiguity. Hurskainen [41] performed a morphological analysis of Swahili using Constraint Grammar Parser while De Pauw et al. [36] used a data-driven approach for morphological analysis of Swahili. The resultant lemma was used to compile a corpus-based dictionary. Early experiments on English–Swahili translation were presented by De Pauw et al. [17] who used GIZA++ to carry out word alignment. Elwell [22] attempted to leverage the Swahili verbal mono-syllabicity morphemes with Naive Bayes algorithm for morphological

analysis. A breakthrough in automating Swahili occurred when the university of Helsinki developed both unannotated and annotated Swahili corpora that have been used for NLP tasks. Ng'ang'a [38] used the corpus to present an automatic lexical acquisition method that learns semantic properties of Swahili words by the self organizing Map algorithm. Earlier, De Pauw et al. [42] had used data-driven taggers for part-of-speech tagging on the annotated Helsinki corpus. It should be noted that most of these works employed the inferior machine learning algorithms when compared to deep learning algorithms [1].

3.2. Deep Learning

Deep learning [5] has been adopted for processing in various fields including sensor drifting because of its robustness [43]. These fields include NLP which has witnessed development from handcrafted features in traditional approaches to machine learning and deep learning techniques. According to Hassan and Mahmood [44], trained linear classifiers and n-gram models treat words as atomic units; these models cannot share parameters and suffer from data sparsity. Artificial neural networks [2] are designed to offer solutions to the limitations of the classical models. Bengio et al. [23] proposed statistical language models where feed-forward neural networks had fixed length context. According to Mikolov et al. [3], statistical models are limited because of the limited context, hence the need for models that can implicitly encode temporal information for contexts with arbitrary length. It is now common for NLP applications to employ deep neural networks [45] to learn word representations using language models [44]. The language models could comprise of recurrent neural networks (RNNs) [46] and/or convolutional neural networks (CNNs) [31].

RNNs are popular with sequential text processing because of their capability to capture and preserve superior and appropriate statistics in a fixed-sized hidden layer. However, the long short term memory (LSTM) [33] was introduced because RNNs only consider recent words. The need to consider past and future information during text processing informed the development of bi-directional long short term memory (Bi-LSTM) [47]. Afterwards, Srivastava et al. [32] proposed the highway networks that have similar memory cells to the LSTM but can allow training of deeper networks by carrying some input directly to the output. Recently, convolutional neural networks have found their way into NLP though they were initially designed for computer vision [48]. This is motivated by the CNN's ability to extract high quality features, leading to CNN models posting significant results in sentiment analysis [49], parsing [50], search query retrieval [51] and part-of-speech tagging [52]. The recent trend is to combine the strengths of the CNN and the RNN to design superior models for NLP [21,44,49,53].

3.3. Word Representation

Learning word embeddings for frequent words is sufficiently handled by word representation models such as word2vec [24] and Glove [54]. However, these models are deficient in representing rare words because they rely on context in the corpus, hence the emergence of compositional models where word representation vectors are generated from sub-word units such as characters, character n-gram, morphemes and syllable-like units. Table 2 provides a summary of these compositional models. The character-aware language models, although effective in handling the rare and new words, inadequately capture semantic meaning of words because characters carry no semantic meaning. The morpheme based models and models using syllable-like units take care of semantic meaning in the word representation but the external segmentation algorithms introduce errors in the models [29]. Our proposed model uses a finite syllabic alphabet to resolve the problem of segmentation errors.

Table 2. The various compositional models for word representation.

Subword	Model	Study
Morphemes	– Morphological RNNs Morph2Vec	Lazaridou et al. [14] Luong et al., 2013 [27] Ustun et al., 2018 [29]
Characters	C2W LSTM-char	Ling et al., 2015 [8] Kim et al., 2016 [11]
Word + Morpheme	Morph-LBL MorphemeCBOW	Cotterell et al., 2015 [55] Qiu et al., 2014 [28]
Syllable-like	– LSTM-syl Subword RNN	Yu et al., 2017 [30] Assylbekov et al., 2017 [21] Mikolov et al., 2012 [9]
n-gram	Charagram Fasttext –	Weiting et al., 2016 [26] Bojanowski et al., 2017 [13] Grave et al., 2018 [56]
Word + Characters	CWE Bi-LSTM/CNN	Chen et al., 2015 [12] Heigold et al., 2017 [57]
Characters + Morphemes	Char2Vec	Cao and Rei, 2016 [15]
Word, characters, Trigrams	Bi-LSTM	Vania and Lopez, 2017 [25]

4. The Proposed Model (WEFSE)

We present WEFSE architecture that comprises of a single layer of convolutional neural network and a highway network. We further outline the syllable-aware language model. Generally, we compose the word representation vectors from syllables such that the representation of a word w is given by:

$$w = f(\mathbf{S}_s, \sigma(s)) \tag{1}$$

where σ is the embedding function that looks up for syllable, stacks and returns a sequence of syllables of a word; $\mathbf{S}_s$ is a parameter matrix representing the vocabulary of syllables; and f is a convolutional function used to compose the word embeddings by taking $\sigma(s)$ and $\mathbf{S}_s$ as input.

4.1. WEFSE

WEFSE uses a convolutional function to compose word representation vectors from the constituent syllable embeddings. As illustrated in Figure 1, the model has a highway network for processing the interaction among the syllables. With a slight modification of the work by Assylbekov et al. [21], a word k that comes in as input is decomposed into a sequence of syllables by looking up the syllables from a finite syllabic alphabet. That is, with the definite syllable alphabet S, the input word k is split into a sequence of syllables $s_1 ... s_l$ where l is the length of k:

$$k = [s_1, ..., s_l]. \tag{2}$$

Given S as a finite vocabulary of syllables and $s_i \in S$, the index of s_i is defined as a one-hot vector $\mathbf{1}_{s_i} \in \mathbb{R}^{|S|\times 1}$, $|S|$ being the size of the syllable vocabulary. We define a projection layer $\mathbf{P}_s \in \mathbb{R}^{d\times|S|}$, where d is the dimension of parameters for each syllable. The embedding for each syllable is therefore obtained as:

$$\mathbf{E}_{s_i} = \mathbf{P}_s.\mathbf{1}_{s_i} \in \mathbb{R}^{d\times 1} \tag{3}$$

The composed word embeddings $\mathbf{E}_w$ from the syllable embeddings $\mathbf{E}_{s_i}$ are given by:

$$\mathbf{E}_w = [\mathbf{E}_{s_1}, ..., \mathbf{E}_{s_l}]. \tag{4}$$

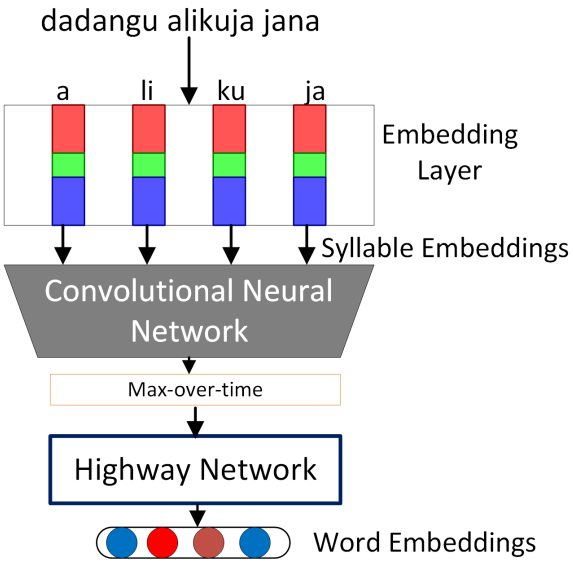

Figure 1. The Architecture of WEFSE demonstrating how the Swahili word *alikuja* is split into its respective syllables and subsequent composition of the embeddings. The convolutional layer provides the composition function to generate the word embeddings.

4.1.1. Convolutional Neural Network

With $s_1....s_l$ being the sequence of syllables of a word k, then the syllable-level representation of k is given by the matrix $\mathbf{S}^k = [\mathbf{E}_{s_1}, .., \mathbf{E}_{s_l}] \in \mathbb{R}^{d \times l}$ where l is the length of word k, ith column corresponds to the syllable embedding for S_i. We then apply a narrow convolution between $\mathbf{S}^k$ and $\mathbf{F} \in \mathbb{R}^{d \times n}$ of width n resulting into a feature map $f^k \in \mathbb{R}^{l-n+1}$ and then add a bias. That is, the ith element of f^k is computed as follows:

$$f^k[i] = tanh(\langle \mathbf{S}^k[*, i : i + n - 1], \mathbf{F} \rangle + b) \tag{5}$$

where $\langle A, B \rangle = Tr(AB^T)$ is the Frobenius inner product while b is the bias. With filters of varying width, we compute max-over-time of each feature map as follows:

$$y^k = max_i f^k[i]. \tag{6}$$

We then derive the word embeddings by concatenating the feature map as follows:

$$w_t = [y_1^k, .., y_h^k] \tag{7}$$

where h is the number of filters applied.

4.1.2. Highway Network

This layer is responsible for processing the interactions between the sequence of syllables. The highway layer allows some dimensions of the word embeddings w_t to be carried or transformed. Typically, the highway layer performs the following:

$$z = t \circ g(\mathbf{W}_H y + b_H) + (1 - t) \circ y \tag{8}$$

where g is a non-linearity, $t = \sigma(\mathbf{W}_T y + b_T)$ is the transform gate and $(1 - t)$ is the carry gate with $\mathbf{W}_T$ and $\mathbf{W}_H$ being square matrices.

4.2. Syllable-Aware Language Model

We adopt Vania and Lopez's [25] version of the language model that uses finite output vocabulary. This enables us to properly compare language models using the perplexity values because they provide similar event spaces. Figure 2 illustrates the architecture of our language model that employs the LSTM, a variant of the recurrent neural network language model of Mikolov et al. [3]. The language model uses the generated word embeddings from syllable embeddings to make predictions of a target word, hence the name syllable-aware. Given a sequence of words $w_{1:t} = w_1, ..., w_t$ and $w_t \in V$ where V is a finite vocabulary set, we compute:

$$P(w_1, ..w_t) = \prod_{t=1}^{T} P(y_t | w_{1:t-1}) \tag{9}$$

where $y_t = w_t$ if w_t is in the output vocabulary and $y_t =$ UNK otherwise. We report the perplexity, that is, the geometry average as:

$$Perplexity = \frac{1}{\prod_{t=1}^{T} P(y_t | w_{1:t-1})} \tag{10}$$

The training in the language model involves minimizing the negative log-likelihood (NLL) of the sequence, which is given by:

$$NLL = -\sum_{t=1}^{T} \log P(y_t | w_{1:t-1}) \tag{11}$$

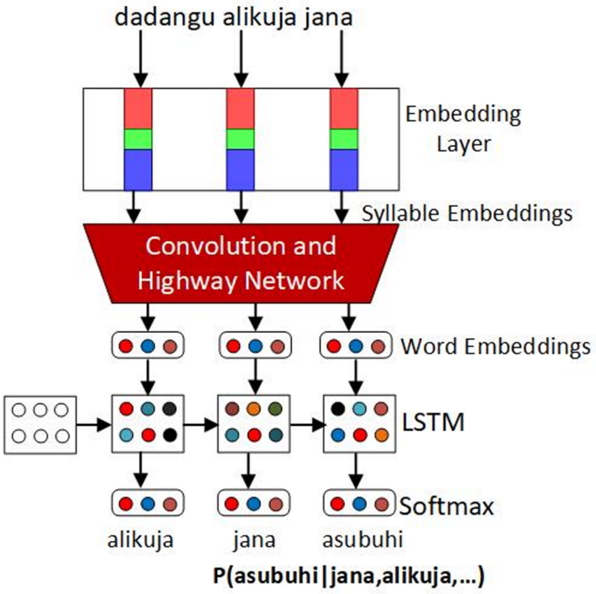

Figure 2. The syllable-aware language model architecture. The model demonstrates prediction of the next word *asubuhi* from its context.

In particular, with the LSTM, once we have generated the word embeddings $w_{1:t}$ for a sequence of words, we produce a sequence of states $h_{1:t}$ and then predict the next word accordingly using the following equations:

$$h_t = LSTM(w_t, h_{t-1}) \tag{12}$$

$$P(y_{t+1} | y_{1:t}) = Softmax(\mathbf{V}^T . h_t) \tag{13}$$

where w_t is the word representation, h_{t-1} is the previous state, y_{t+1} is the predicted target word, $y_{1:t}$ are the context words and $\mathbf{V}$ is the weight matrix.

Note that the softmax is a d × V table, which encodes the likelihood of every word type in a given context and V is from a finite output vocabulary. The LSTM uses $x_t = w_t, h_{t-1}, c_{t-1}$ to determine h_t as follows:

$$\begin{aligned} i_t &= \sigma(\mathbf{W}^i x_t + \mathbf{U}^i h_{t-1} + b^i) \\ f_t &= \sigma(\mathbf{W}^f x_t + \mathbf{U}^f h_{t-1} + b^f) \\ o_t &= \sigma(\mathbf{W}^o x_t + \mathbf{U}^o h_{t-1} + b^o) \\ g_t &= tanh(\mathbf{W}^g x_t + \mathbf{U}^g h_{t-1} + b^g) \\ c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\ h_t &= o_t \circ tanh(c_t) \end{aligned} \tag{14}$$

where σ, $tanh$ and $\circ$ are element-wise sigmoid, hyperbolic tangent and multiplication functions, respectively. i_t, f_t and o_t are the input, forget and output gates, respectively.

5. Experiments

We carried out various experiments to demonstrate the quality of the word embeddings from syllable embeddings. We hereby describe the experimental setup, datasets, baselines and training employed.

5.1. Datasets

We used Swahili, Xhosa and Shona datasets in the experiments. For Swahili, we used data collected from online newspapers by Gelas et al. [34], which contain 28 million unique words whose scope includes sports, general news, religion and family life. The dataset was already partitioned into training, development and test data. However, following Kim et al.'s [11] approach, we created small and medium datasets containing 28,000 and 514,000 unique words, respectively, for appropriate comparison with other prior experiments. We obtained the small dataset by partitioning the provided training data into train, development and test sets in the ratio 70:10:20, respectively, using a text editor. For the medium dataset, we used the training, development and test data as provided by the authors. The raw Xhosa https://github.com/godlytalias/Bible-Database/tree/master/Xhosa and Shona https://github.com/teusbenschop/shona data were collected from respective online religious materials, which limited the context of the data. This was meant to test the effect of the scope of data to the quality of the embeddings [56]. We also partitioned the Xhosa and Shona datasets following the method applied to obtain the small Swahili dataset. We then preprocessed Swahili, Xhosa and Shona datasets by lowercasing, removing punctuation and incorporating the start and end of sentence markers, as well as replacing singletons with <unk> for uniformity with prior works, although syllables are capable of handling out of vocabulary words. Table 3 provides the summary of the datasets.

Table 3. The various datasets used in this study. The partition sizes are presented as k and M for thousand and million words, respectively.

Dataset	Unique Words	Train	Development	Test
Swahili (small)	28 k	6.84 M	970 k	2 M
Swahili (medium)	514 k	204 M	1.8 M	1.09 M
Xhosa	24 k	2.25 M	321 k	642 k
Shona	22 k	2.7 M	385 k	771 k

5.2. Baselines

We benchmarked the quality of WEFSE on the works of Assylbekov et al. [21] and Vania and Lopez [25]. Assylbekov et al. [21] carried out experiments using various models (sum, concatenate and convolutional neural network) to demonstrate the performance of syllable-like units in various models.

We trained Swahili on these models using both the applied external hyphenator algorithm and our finite syllabic alphabet. These experiments informed us on the model to adopt for our composition function. We used the work of Vania and Lopez [25] to compare how syllables from a finite alphabet could perform against characters and character trigrams. In addition to this, we tested whether the syllables could post good results when bi-directional LSTM is used as the composition function; we report the results of the baseline experiments in Table 4.

5.3. Experimental Setup

We carried out our experiments using the Tensorflow framework with the dimensions of character, syllable and word embeddings set to 200, 200 and 300, respectively. The baseline experiments adopted the experiment settings of the respective works. The various datasets were placed in respective sub-folders and stored in a common folder named data. Similar to Vania and Lopez [25] and for comparison purposes, we set the maximum output vocabulary to most frequent 5000 training words. We implemented the LSTM with hidden units of size 200 and adopted the settings of Assylbekov et al. [21] for the convolutional neural and highway networks; the details are provided in the Appendix A.

5.4. Training

The choices made for optimization were guided by the works of Zaremba et al., Assylbekov et al., Kim et al., and Vania and Lopez [11,21,25,58]. We applied varying optimization settings for the small and medium Swahili datasets with the small dataset running a batch size of 20 for 25 epochs and the medium dataset using 100 and 15 for batch size and number of epochs, respectively. For the convolutional layer, the features per width and convolutional dimension were set the same as Assylbekov et al. [21]. We trained by truncated back propagation through time, propagating at 35 time steps using stochastic gradient descent [59]. The learning rate began at 1.0 and was reduced to half if the validation perplexity did not decrease by 0.1 after three epochs [25]. Following Kim et al. [11], we randomly initialized the model parameters over a uniform distribution $[-0.05, 0.05]$ and applied regularization through dropout with probability of 0.5 on the LSTM input-to-hidden layers and hidden-to-output softmax layer. However, we did not use dropout on the initial highway to the LSTM layer.

6. Results and Discussion

In this section we report and explain the results from the baseline experiments and our model. The quality of the word embeddings is informed by low perplexity values of the language models.

6.1. Baseline Results

We performed two baseline experiments to assess the quality of WEFSE. Assylbekov et al. [21] composed word embeddings from syllable-like units using sum, concatenation (concat) and convolutional neural network (cnn) models. These models provided us with a platform to assess our word embeddings (WEFSE) by comparing the quality of word embeddings generated from finite syllables and syllable-like units. Therefore, we ran the small Swahili dataset on the models using syllables from both the syllabic alphabet and an external hyphenator. The results of these experiments are reported in Table 4 where the LSTM-syl (cnn) model outperforms the LSTM-syl (sum) and LSTM-syl (concat) models for both the syllables from the hyphenator and our syllables. This is attributed to the ability of convolutional neural networks to extract quality syllable features that are used to compose the word embeddings. However, the syllable-like units outperform our syllables across the models though the results are competitive to state-of-art models. Note the number of unique syllables in the scenarios; the hyphenator generates **18,440** unique syllables compared to **210** from our syllabic alphabet for the same dataset. This demonstrates that syllable-like units are random combinations of characters that do not carry any semantic meaning. According to Ustun et al. [29], quality embeddings

are composed from meaningful sub-word units. We resolved to use the CNN for our model based on this baseline experiment.

The work of Vania and Lopez [25], in which words, characters, character trigrams and morphemes are used to compose word embeddings using bi-directional LSTM and add functions, offered another benchmark for our WEFSE. However, we did not experiment on the morphemes or the add function because of the need for an external segmenter and the earlier baseline experiment which had used sum function. The objective was to assess our WEFSE on a bi-directional LSTM as a composition function and compare the performance of syllables to characters and trigrams. Again, we carried out the experiments using the small Swahili dataset and the results are outlined in Table 4. The syllable model outperforms the character-aware counterpart negating the conclusion by Assylbekov et al. [21] that characters are superior to syllables. However, the trigrams outperform both character-aware and syllable-aware models, even though they possess no semantic meaning. Generally, the two baselines experiments demonstrated the superiority of CNNs over bi-directional LSTMs in the extraction of quality features for NLP tasks. However, a compositional model combining the two functions is worth exploring and we leave it for future consideration.

Table 4. The baseline results for the LSTM-syl [21] and Bi-LSTM [25]. Our Model (WEFSE) posts very competitive results to state-of-art on the LSTM-syl (cnn + WEFSE) model. The unit of measure is perplexity; the lower is the value, the better is the representation. The bolded values represent the best performance.

Model	Valid	Test	Unique Syllables
LSTM-syl (sum)	39.082	39.082	18,440
LSTM-syl (concat)	59.399	63.625	18,440
LSTM-syl (cnn)	37.509	38.935	18,440
Bi-LSTM (char)	68.968	70.976	–
Bi-LSTM (trigram)	62.181	64.548	–
LSTM-syl (sum + WEFSE)	54.189	58.189	210
LSTM-syl (concat + WEFSE)	87.546	95.906	210
LSTM-syl (cnn + WEFSE)	**43.567**	**45.867**	210
Bi-LSTM (WEFSE)	63.196	65.548	213

Generally, we argue that our model is better than the baselines in representing Swahili words. This is because we are convinced that proper representation of Swahili words should effectively cater for its agglutinative features, especially in the verbs that may consist of subject, tense, relative, root and some extensions, which are themselves syllables. This implies that a good compositional model for Swahili should split the words into sub-word units without distorting the agglutinative components in terms of orthography and position. Table 5 provides a comparative analysis on how the compositional models were splitting Swahili words. It clearly demonstrates that syllable models maintain the agglutinative components. We therefore expect superior word representation vectors from syllables as compared to characters and trigrams.

Table 5. Comparative analysis of how Swahili words are split when characters, trigram and syllables are used as compositional components. The first two are verbs with subject prefix, tense prefix and root while the last two are noun and pronoun, respectively.

Word	Characters	Trigrams	Syllables
walikuja	w, a, l, i, k, u, j, a	wal, ali, lik, kuj, uja, walikuja	wa, li, ku, ja
atalipa	a, t, a, l, i, p, a	ata, tal, ali, ipa, atalipa	a, ta, li, pa
mchezo	m, c, h, e, z, o	mch, che, hez, ezo, mchezo	m, che, zo
mimi	m, i, m, i	mim, imi, mi, mimi	mi, mi

6.2. Our Model Results (WEFSE)

We present the results of our syllable-aware language model in Table 6. The experiments were carried out on Swahili, Xhosa and Shona, which are agglutinative and syllabic-based languages.

We observed unexpected results on the Shona and Xhosa datasets where the character model slightly outperforms the syllable model. We attribute these results to the limited context of the datasets and the fact that Swahili is morphologically richer than Xhosa and Shona [60]. For Swahili, the syllable model achieves better results than the character model on both the small and medium datasets. The perplexity values achieved by this model compete favorably with state-of-the-art models in NLP. The results on the medium dataset allude to the fact that rare words are well handled when the word embeddings are composed from syllable embeddings. Indeed, the model represents the words better because the syllabic alphabet supplies the constituent syllables of any word for the model to learn from. Therefore, these results validate the quality of WEFSE and our hypothesis.

It should be noted that the syllable vocabulary sizes are different because we used unique syllabic alphabet for Swahili, Xhosa and Shona. Xhosa has a large syllabic vocabulary size that employs the entire English alphabet. During processing, we added two characters to encapsulate each word, padding character, and start and end of sentence markers to the syllable dictionary. This explains the size of Swahili vocabulary.

Table 6. The perplexity results (PPL) of the WEFSE model. The Syllable model completely outperforms the character counterpart on both small and medium datasets. The bolded values represent the best performance.

Dataset	Character		Syllable	
	Size	PPL	Size	PPL
Xhosa	37	114.127	272	114.618
Shona	33	110.167	235	110.242
Swahili (small)	33	31.576	213	**31.229**
Swahili (medium)	33	47.893	219	**45.859**

Further, we investigated the effect of different character and syllable embedding dimensions on the perplexity values. Figure 3 shows that there is no statistically significant effect of the dimension on perplexity values. However, the most appropriate embedding dimensions for our experiment could have been 200 or 500 for both characters and syllables; we used a dimension of 200.

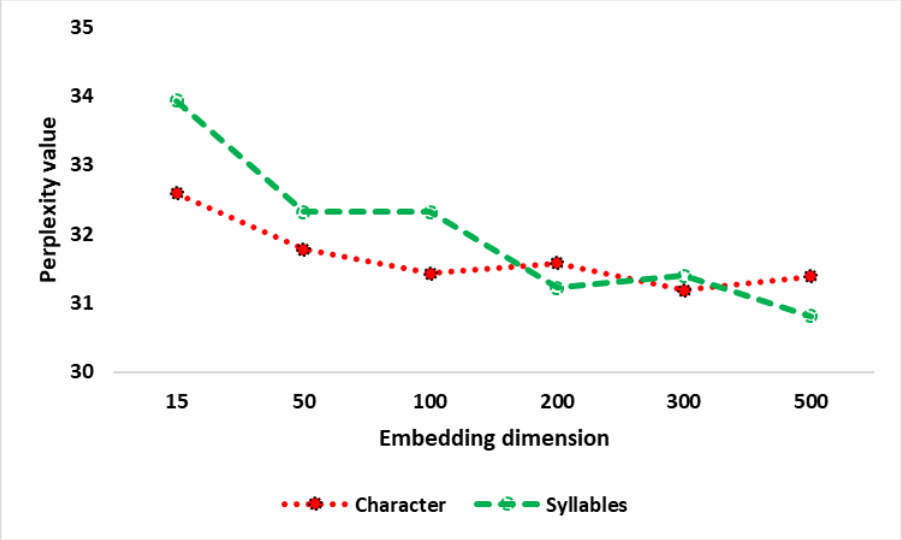

Figure 3. The effect of different character and syllable dimensions on perplexity values. The most appropriate dimension for our experiment is 200 or 500.

Communication is critical in every sector of a community, society, region, country or continent. This is because communication supports every field including health, education, trade, engineering, industry, transport, service, hospitality and games. Therefore, automated information systems which are facilitated by NLP play an important role in economic development of a region. We are of the opinion that our WEFSE will inspire development of such systems for East and Central Africa to foster and enhance regional integration and development.

6.2.1. Qualitative Analysis

We also used the nearest neighbor test to explore the quality of the WEFSE compared to word2vec by Mikolov et al. [24], where we considered the cosine similarity of the Top 5 words for in-vocabulary (*wanakutana* and *mchezo*) and nonce words (*mulinga* and *kusorta*) in the small Swahili test file. The results in Table 7 attest to the ability of our word embeddings to associate similar word. WEFSE processed and associated words with the out-of-vocabulary (nonce) words as opposed to word2vec which returned an error. Indeed, WEFSE associated syntactically and semantically similar words with *wanakutana* being associated with present tense verbs while *mchezo* with nouns. This suggests that the word representation vectors from WEFSE are superior because they exploit the composition aspect (syllables) and context of the words. The results demonstrate that words in agglutinative and syllabic-based languages are inadequately represented using word vectors. These results suggest that constructing word embeddings from syllable embeddings is useful in deep learning when training on data that can benefit from contextual information, especially on NLP tasks such as machine translation (common in automated systems using speech), sentiment analysis, parsing and question and answer systems.

Table 7. The syntactically and semantically similar words using nearest neighbors.

Model	Wanakutana	Mchezo	Mulinga	Kusorta
word2vec	italeta	mtoto	-	-
	amri	aliwataka	-	-
	frank	upande	-	-
	kujihusisha	mawaziri	-	-
	waweze	kiongozi	-	-
WEFSE	wanasubiri	mchele	mulinda	kusota
	wanaamini	mchepuo	mulanda	kusoma
	wanatakiwa	mchakato	muhimu	kusomea
	wanachama	mchango	mulla	kusomewa
	wananchi	mchaka	musa	kusonga

6.2.2. Word Analogy Task

We further evaluated the WEFSE using the word analogy test. This test uses a triplet of words with the goal being to guess the fourth word such that given A is to B (A:B) and C is to D (C:D). In this case, the vectors of A, B and C are used to evaluate the vector of D. The vector of D would be computed by $X_B - X_A + X_C$ where X_B, X_A and X_C are the vectors of words B, A and C, respectively. A typical example of a word analogy question is "Boy:Girl::Man:?". The average accuracy over the entire corpus provides the performance measure for the analogy.

The non-existence of a Swahili word analogy dataset forced us to develop a dataset based on the English dataset introduced by Mikolov et al. [24]. Therefore, because there is no direct translation for some categories such as comparatives, superlatives, city-in-state and antonyms, we removed or replaced them accordingly. For example, we added a new category for sounds (*tanakali*) and replaced the cities with constituencies and their corresponding counties in Kenya. The resultant analogy dataset has **12,864** questions on which we experimented using word2vec and our WEFSE. The results in Table 8 show that our model outperforms the word2vec model, which is based on words. This demonstrates that word embeddings generated from syllables embeddings are better than word-based embeddings.

Table 8. The word analogy results for WEFSE and word2vec. Our model outperforms the word-based word representation. The bolded values represent the best performance.

Model	Accuracy (%)
word2vec [24]	15.5
WEFSE (Ours)	**64.8**

7. Conclusions and Future Work

We presented the Swahili syllabic alphabet and used it to generate word representation vectors. The study employed a convolutional neural network and a highway network to compose WEFSE. We demonstrated the quality of the word embeddings with the syllable-aware language model, achieving very competitive perplexity values that outperformed the character-aware counterpart on both small and medium datasets. We also confirmed the quality of WEFSE using the word analogy task after developing the Swahili analogy dataset. The performance of downstream NLP tasks depends on the quality of word representation. We therefore propose using WEFSE on part-of-speech tagging, machine translation and text classification tasks in future works. In addition, we will explore combining the strength of CNNs and RNNs in modeling the composition function for generating word embeddings from syllable embeddings. The need for a generic syllabic alphabet for all Bantu languages is not far-fetched and will be part of our future considerations.

Author Contributions: C.S.S. conceptualization, methodology, software, writing original draft, review and editing, Z.S. funding acquisition, resources, supervision, L.Q. supervision, resources, R.M. software, review and editing.

Funding: This research was supported by the Sichuan Science and Technology Program under grants 2018FZ0097 and 2018GZDZX0006.

Acknowledgments: Casper Shikali is sponsored by Chinese Government Scholarship. He also appreciates the management of South Eastern Kenya University, led by Geoffrey Muluvi, for granting him study leave to undertake this study. We thank all the anonymous reviewers for their very useful comments and discussion.

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

Abbreviations

The following abbreviations are used in this manuscript:

NLP	Natural Language Processing
WEFSE	Word Embeddings From Syllable Embeddings
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
RNN	Recurrent Neural Network
RNNs	Recurrent Neural Networks
LSTM	Long Short Term Memory
Bi-LSTM	Bi-directional Long Short Term Memory

Appendix A

The model used tanh and ReLU as the non-linear activation functions in the convolutional and highway layers, respectively. For the small dataset, we set the CNN output dimension as $(1 + 2 + 3) \times 60$ and feature per width as 60. For the medium dataset, the CNN output dimension was $(1 + 2 + 3) \times 195$ and feature per width was 195. Generally, the convolutional filter widths were [1, ..., L] where L = 2, 3, 4 was experimented. We ensured that the highway layer dimension was equal to the CNN output dimension.

During training, the same as Assylbekov et al. [21], we propagated at 35 time steps using stochastic gradient descent on both the small and medium datasets. The learning rate was initialized to 1.0 and reduced by half if the validation perplexity value did not decrease by 0.1 after three epochs [25].

We trained for 25 and 15 epochs on the small and medium datasets, respectively. A 0.5 probability dropout was used for regularization [59] with model parameters initialized randomly using uniform distribution [−0.05, 0.05].

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. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
3. Mikolov, T.; Karafiát, M.; Burget, L.; Černockỳ, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010.
4. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: Cambridge, MA, USA, 2013; pp. 3111–3119.
5. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
6. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
7. Botha, J.; Blunsom, P. Compositional morphology for word representations and language modelling. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 January 2014; pp. 1899–1907.
8. Ling, W.; Dyer, C.; Black, A.W.; Trancoso, I.; Fermandez, R.; Amir, S.; Marujo, L.; Luís, T. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 1520–1530. [CrossRef]
9. Mikolov, T.; Sutskever, I.; Deoras, A.; Le, H.S.; Kombrink, S.; Cernocky, J. *Subword lAnguage Modeling with Neural Networks*; Faculty of Information Technology, Brno University of Technology: Brno, Czech Republic, 2012.
10. Turian, J.; Ratinov, L.; Bengio, Y. Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 384–394.
11. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A.M. Character-aware neural language models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
12. Chen, X.; Xu, L.; Liu, Z.; Sun, M.; Luan, H. Joint learning of character and word embeddings. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
13. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
14. Lazaridou, A.; Marelli, M.; Zamparelli, R.; Baroni, M. Compositional-ly derived representations of morphologically complex words in distributional semantics. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, 4–9 August 2013; Volume 1, pp. 1517–1526.
15. Cao, K.; Rei, M. A Joint Model for Word Embedding and Word Morphology. In Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, 11 August 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 18–26. [CrossRef]
16. Thomas, S.; Seltzer, M.L.; Church, K.; Hermansky, H. Deep neural network features and semi-supervised training for low resource speech recognition. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6704–6708. [CrossRef]

17. De Pauw, G.; Wagacha, P.W.; de Schryver, G.M. *Towards English-Swahili Machine Translation*; Research Workshop of the Israel Science Foundation: Tel Aviv-Yafo, Israel, 2011.
18. Williams, E. *Bridges and Barriers: Language in African Education and Development*; Routledge: Abingdon-on-Thames, UK, 2014.
19. Choge, S.C. A Morphological Classification of Kiswahili. *Kiswahili* **2018**, *80*, 1.
20. Indakwa, J.; Ballali, D. *Beginner's Swahili*; Hippocrene Books: New York, NY, USA, 1995.
21. Assylbekov, Z.; Takhanov, R.; Myrzakhmetov, B.; Washington, J.N. Syllable-aware Neural Language Models: A Failure to Beat Character-aware Ones. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 1866–1872. [CrossRef]
22. Elwell, R. Using Syllables as Features in Morpheme Tagging in Swahili. In Proceedings of the Fifth Midwest Computational Linguistics Colloquium, East Lansing, MI, USA, 5 May 2008.
23. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
24. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, AZ, USA, 2–4 May 2013.
25. Vania, C.; Lopez, A. From Characters to Words to in Between: Do We Capture Morphology? In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 2016–2027. [CrossRef]
26. Wieting, J.; Bansal, M.; Gimpel, K.; Livescu, K. Charagram: Embedding Words and Sentences via Character n-grams. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1504–1515. [CrossRef]
27. Luong, T.; Socher, R.; Manning, C. Better word representations with recursive neural networks for morphology. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning, Sofia, Bulgaria, 8–9 August 2013; pp. 104–113.
28. Qiu, S.; Cui, Q.; Bian, J.; Gao, B.; Liu, T.Y. Co-learning of word representations and morpheme representations. In Proceedings of the 25th International Conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014; pp. 141–150.
29. Üstün, A.; Kurfalı, M.; Can, B. Characters or Morphemes: How to Represent Words? In Proceedings of the Third Workshop on Representation Learning for NLP, Melbourne, Australia, 20 July 2018; pp. 144–153.
30. Yu, S.; Kulkarni, N.; Lee, H.; Kim, J. Syllable-level Neural Language Model for Agglutinative Language. In Proceedings of the First Workshop on Subword and Character Level Models in NLP, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 92–96. [CrossRef]
31. LeCun, Y.; Bengio, Y. *Convolutional Networks for Images, Speech, and Time Series*; The Handbook of Brain Theory and Neural Networks: Cambridge, MA, USA, 1995.
32. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway networks. In Proceedings of the 2015 Deep Learning Workshop International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1504–1515.
33. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
34. Gelas, H.; Besacier, L.; Pellegrino, F. *Developments of Swahili Resources for An Automatic Speech Recognition System*; SLTU—Workshop on Spoken Language Technologies for Under-Resourced Languages: Cape-Town, South Africa, 2012.
35. Polomé, E.C. *Swahili Language Handbook*; Center for Applied Linguistics: Washington, DC, USA, 1967.
36. De Pauw, G.; De Schryver, G.M. Improving the computational morphological analysis of a Swahili corpus for lexicographic purposes. *Lexikos* **2008**. [CrossRef]
37. Deen, K.U.D.S.U. The Acquisition of Nairobi Swahili: The Morphosyntax of Inflectional Prefixes and Subjects. Ph.D. Thesis, University of California, Los Angeles, CA, USA, 2002.

38. Ng'ang'a, W. Semantic analysis of Kiswahili words using the self organizing Map. *Nord. J. Afr. Stud.* **2003**, *12*, 407–425.
39. Amidu, A.A. Kiswahili: People, language, literature and lingua franca. *Nord. J. Afr. Stud.* **1995**, *4*, 104–123.
40. Masengo, I.J. Cross-Linguistic Influence in Third Language Production Among Kiswahili Learners. Ph.D. Thesis, Makerere University, Kampala, Uganda, 2018.
41. Hurskainen, A. Disambiguation of morphological analysis in Bantu languages. In Proceedings of the 16th Conference on Computational Linguistics-Volume 1, Copenhagen, Denmark, 5–9 August 1996; Association for Computational Linguistics: Stroudsburg, PA, USA, 1996, pp. 568–573.
42. De Pauw, G.; De Schryver, G.M.; Wagacha, P.W. Data-driven part-of-speech tagging of Kiswahili. In Proceedings of the International Conference on Text, Speech and Dialogue, Brno, Czech Republic, 11–15 September 2006; pp. 197–204.
43. Liu, Q.; Hu, X.; Ye, M.; Cheng, X.; Li, F. Gas recognition under sensor drift by using deep learning. *Int. J. Intell. Syst.* **2015**, *30*, 907–922. [CrossRef]
44. Hassan, A.; Mahmood, A. Convolutional recurrent deep learning model for sentence classification. *IEEE Access* **2018**, *6*, 13949–13957. [CrossRef]
45. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Kingsbury, B.; et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
46. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
47. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
48. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
49. Du, J.; Gui, L.; He, Y.; Xu, R.; Wang, X. Convolution-Based Neural Attention with Applications to Sentiment Classification. *IEEE Access* **2019**, *7*, 27983–27992. [CrossRef]
50. Yih, W.T.; He, X.; Meek, C. Semantic parsing for single-relation question answering. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; Volume 2, pp. 643–648.
51. Shen, Y.; He, X.; Gao, J.; Deng, L.; Mesnil, G. Learning semantic representations using convolutional neural networks for web search. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 7–11 April 2014; pp. 373–374.
52. Santos, C.D.; Zadrozny, B. Learning character-level representations for part-of-speech tagging. In Proceedings of the 31st International Conference on Machine Learning (ICML 2014), Beijing, China, 21–26 June 2014.
53. Xiao, Y.; Cho, K. Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers. *arXiv* **2016**, arXiv:1602.00367.
54. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1532–1543.
55. Cotterell, R.; Schütze, H. Morphological word-embeddings. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 1287–1292.
56. Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; Mikolov, T. Learning Word Vectors for 157 Languages. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan, 7–12 May 2018; European Languages Resources Association (ELRA): Paris, France, 2018; pp. 3483–3487.
57. Heigold, G.; Neumann, G.; van Genabith, J. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Association for Computational Linguistics, Proceedings of the 15th Conference, Valencia, Spain, 3–7 April 2017*; European Chapter, Long Papers; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; Volume 1, pp. 505–513.
58. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. *arXiv* **2014**, arXiv:1409.2329.

59. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2015; pp. 2377–2385.
60. Bostoen, K. Bantu spirantization: Morphologization, lexicalization and historical classification. *Diachronica* **2008**, *25*, 299–356.