**Deep Learning Innovations in Nyishi Language Processing: A Focus on Word Sense Disambiguation**

This research methodology focuses on developing a deep learning-based Word Sense Disambiguation (WSD) system for the Nyishi language.

1. **Data Collection and Preparation**

The primary dataset for this research consists of a manually curated Nyishi language. Thirty-six item tag sets have been created, each of which includes 28 distinct letters, 18 distinct consonants, 7 distinct vowels, 2 clusters, and 1 glottal for the Nyishi POS system. Additionally, more than 25000 entries from the Nyishi-to-English Dictionary have been collected, and this was done both manually and automatically.

1. **Data Preprocessing**

To prepare the data for input into deep learning models, several preprocessing steps are applied:

* **Tokenization:** Each sentence is tokenized into individual words and characters, aligning with the Nyishi-specific tag set.
* **Part-of-Speech (POS) Tagging:** POS tagging is used to assign syntactic roles to each word, which aids in identifying the grammatical context that influences word meanings.
* **Tag Set Alignment:** The unique thirty-six-item tag set is integrated into the preprocessing pipeline to standardize the linguistic representation of Nyishi words. This ensures that each token is enriched with detailed linguistic features, improving the model’s ability to disambiguate word senses.

1. **Feature Extraction and Selection**

Feature extraction and selection is crucial in training deep learning models to identify the correct word sense. The following techniques are used:

* **Word Embeddings**:
* **Word2Vec**: A pre-trained embedding method that captures the semantic similarity between words based on their context in large corpora. Word2Vec helps represent Nyishi words in a high-dimensional space, making it easier for models to capture semantic nuances.
* **FastText**: Given the rich morphology of the Nyishi language, FastText embeddings are used, which include subword information. This technique is more effective in handling rare words or words that were not seen during training.
* **POS Tag Features**: Adding POS tags as additional input features helps the model understand the grammatical role of a word, aiding in disambiguation.

1. **Data Splitting**

The dataset is divided into two parts:

* **Training Set:** 80% of the dataset is used for training the model, where the model learns patterns and contextual clues for word sense disambiguation.
* **Testing Set:** 20% of the dataset is reserved for evaluating the model's performance. This ensures that the model is tested on unseen data, providing a reliable measure of its generalization ability.

1. **Model building**

The following deep learning models are selected for developing the WSD system:

* **Bidirectional LSTM (BiLSTM)**

BiLSTM networks, which process input in both forward and backward directions, are used to capture the context surrounding a word in both directions. This is particularly helpful in languages like Nyishi, where both the preceding and following words may influence the word's meaning.

* **BERT**

Bidirectional Encoder Representations from Transformers (BERT) is selected for its ability to handle deep contextual information and bidirectional encoding. BERT will be fine-tuned on the Nyishi dataset, taking advantage of its pre-trained language model and adapting it to the specific linguistic characteristics of Nyishi.

* **Hybrid CNN-BiLSTM Models**

A hybrid architecture combining Convolutional Neural Networks (CNN) and BiLSTM is applied to leverage both local context (via CNN) and global dependencies (via BiLSTM). CNNs capture local patterns and are particularly useful for processing fixed-length word clusters, while BiLSTM helps in processing the full sentence context.

1. **Model Evaluation**

The performance of the WSD models is evaluated using the following metrics: Accuracy, Precision, Recall, F1 score, Confusion Matrix and Cross Validation.

1. **Sense Tagged Data**

Words are manually tagged with appropriate senses by linguistic experts. These sense-tagged annotations serve as the target output for training models, enabling effective word sense disambiguation.

Data Collection

Data Preprocessing

Feature Extraction and Selection

Word Embeddings

Word2Vec

Fast Text

POS Tag

Training

Data Splitting

Testing

Model Building

CNN

Bi-LSTM

BERT

CNN-BiLSTM

Model Evaluation

Sense Tagged Test data

Dictionary

Removal of Punctuation &

Abbreviations.

Tokenization

POS Tag

**Proposed Algorithm**

**Step 1: Data Collection**

Let:

* T be the total number of sentences.
* represent the sentences in the dataset.
* Each sentence where is the i-th word in the t-th sentence, and nt is the length of sentence xt​.
* be the POS tag of word .

**Step 2: Data Preprocessing**

**Tokenization:**

Each sentence xt​ is tokenized into words:

**POS Tagging:**

For each word ​, assign a POS tag:

**Tag Set Alignment:**

Each word ​ is mapped to a tag from the Nyishi tag set C:

**Step 3: Feature Extraction and Selection**

**Word Embeddings:**

Using Word2Vec and FastText to represent words in high-dimensional vectors:

* **Word2Vec:** Let be the Word2Vec embedding for word ​.

where d is the dimensionality of the embedding.

* **FastText:** Let be the FastText embedding, including subword information:

**POS Tag Features:**

POS tag is one-hot encoded as:

The final feature vector for each word :

where ⊕ denotes concatenation.

**Step 4: Data Splitting**

The dataset X is split into training and testing sets.

**Step 5: Model Building**

**BiLSTM Model:**

The BiLSTM processes words in both forward and backward directions:

and

The final hidden state:

**BERT Model:**

BERT uses the input sequence of word embeddings:

The output from BERT is the contextualized word representations.

**Hybrid CNN-BiLSTM:**

For each sentence Wt​, CNN applies convolution filters to the embeddings:

The output is fed into BiLSTM:

**Step 7: Model Evaluation**

The performance of the WSD models is evaluated using the following metrics: Accuracy, Precision, Recall, F1 score, Confusion Matrix and Cross Validation.

**Step 8: Sense-Tagged Data**

Let be the set of manually sense-tagged labels. Each word ​ is assigned a sense ​.

**Techniques Used**

* **CNN**

CNN is employed to handle inputs with grid-like topology. As illustrated in Figure 1, CNNs consist of one or more convolutional layers and are primarily used for tasks like disease detection, classification, prediction and other related data-processing applications.  Convolution is the method of slipping a filter across an input signal [[[1]](#footnote-1)] that allows CNNs to analyze the surroundings of a function to present recovered or more precise estimates of its result [[[2]](#footnote-2)]. Additionally, these are well-suited to common functional signal processing and segmentation tasks, making them well-suited for early myopia detection.

A diagram of different types of shapes

Description automatically generated

Figure 1. CNN [[[3]](#footnote-3)]

* **BERT**

BERT is a deep learning model that has revolutionized natural language processing tasks, including word sense disambiguation [[[4]](#footnote-4)]. In the context of Nyishi language processing, BERT can effectively capture the contextual meaning of words by analyzing their relationships in both directions within a sentence [[[5]](#footnote-5)]. This bidirectional approach allows BERT to understand the nuances of polysemous words (words with multiple meanings) based on their surrounding context, making it a powerful tool for resolving ambiguities in the Nyishi language [[[6]](#footnote-6)]. The structure of the BERT pre-training model is shown in Figure 2.

A diagram of a algorithm

Description automatically generated

**Figure 2:** BERT model architecture [[[7]](#footnote-7)]

* **Bi-LSTM**

Bi-LSTM is a deep learning technique that enhances language processing by capturing both forward and backward contextual information in a sentence [[[8]](#footnote-8)]. In the context of Nyishi language processing, particularly for WSD, Bi-LSTM is crucial as it helps model complex relationships between words by considering the preceding and succeeding words simultaneously [[[9]](#footnote-9)]. This bidirectional approach improves the system's ability to correctly identify the meaning of a word based on its context, addressing the challenges posed by the polysynthetic and agglutinative nature of the Nyishi language [[[10]](#footnote-10)].

* **CNN-BiLSTM**

CNN-Bi-LSTM is a hybrid deep learning model that combines CNN and Bi-LSTM networks, making it effective for tasks like WSD in language processing [[[11]](#footnote-11)]. The CNN component captures local contextual information by applying filters to input text, while the Bi-LSTM leverages the temporal dependencies in both forward and backward directions, enabling it to understand the context surrounding a word. Together, CNN-Bi-LSTM can handle the nuances of polysemous words, making it well-suited for the Nyishi language's word sense disambiguation challenges [[[12]](#footnote-12)].

1. Khudayer, Baidaa Hamza, Mohammed Anbar, Sabri M. Hanshi, and Tat-Chee Wan. "Efficient route discovery and link failure detection mechanisms for source routing protocol in mobile ad-hoc networks." IEEE Access 8 (2020): 24019-24032. [↑](#footnote-ref-1)
2. Lu, Renjie. "Malware detection with lstm using opcode language." arXiv preprint arXiv:1906.04593 (2019). [↑](#footnote-ref-2)
3. Rinish Reddy, R., Sadhwika Rachamalla, Mohamed Sirajudeen Yoosuf, and G. R. Anil. "Convolutional Neural Network Based Intrusion Detection System and Predicting the DDoS Attack." In *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2022*, pp. 81-94. Singapore: Springer Nature Singapore, 2022. [↑](#footnote-ref-3)
4. Yap, Boon Peng, Andrew Koh, and Eng Siong Chng. "Adapting BERT for word sense disambiguation with gloss selection objective and example sentences." *arXiv preprint arXiv:2009.11795* (2020). [↑](#footnote-ref-4)
5. Saha, Anik, Alex Gittens, and Bulent Yener. "Word Sense Induction with Knowledge Distillation from BERT." *arXiv preprint arXiv:2304.10642* (2023). [↑](#footnote-ref-5)
6. Du, Jiaju, Fanchao Qi, and Maosong Sun. "Using BERT for word sense disambiguation." *arXiv preprint arXiv:1909.08358* (2019). [↑](#footnote-ref-6)
7. Zhou, Gang. "A Study of the Combination of Semantic Understanding Enhancement Methods and Deep Learning Techniques in English Translation." *Applied Mathematics and Nonlinear Sciences* 9, no. 1. [↑](#footnote-ref-7)
8. Li, Z. H. I., F. A. N. Yang, and Yaoru Luo. "Context embedding based on bi-LSTM in semi-supervised biomedical word sense disambiguation." *IEEE Access* 7 (2019): 72928-72935. [↑](#footnote-ref-8)
9. Shivachi, Casper Shikali, Refuoe Mokhosi, Zhou Shijie, and Liu Qihe. "Learning syllables using Conv-LSTM model for Swahili word representation and part-of-speech Tagging." *Transactions on Asian and Low-Resource Language Information Processing* 20, no. 4 (2021): 1-25. [↑](#footnote-ref-9)
10. Lai, Huei-Ling, Hsiao-Ling Hsu, Jyi-Shane Liu, Chia-Hung Lin, and Yanhong Chen. "Supervised Word Sense Disambiguation on Polysemy with Neural Network Models: A Case Study of BUN in Taiwan Hakka." *International Journal of Asian Language Processing* 30, no. 03 (2020): 2050011. [↑](#footnote-ref-10)
11. Singh, Gurpreet, and C. P. Kamboj. "Hybrid CNN-LSTM Architecture for Bilingual Next-Word Prediction in Punjabi-English Social Media Texts." In *2023 4th IEEE Global Conference for Advancement in Technology (GCAT)*, pp. 1-7. IEEE, 2023. [↑](#footnote-ref-11)
12. Rehman, Anwar Ur, Ahmad Kamran Malik, Basit Raza, and Waqar Ali. "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis." *Multimedia Tools and Applications* 78 (2019): 26597-26613. [↑](#footnote-ref-12)