CS5246 Project: Simplify News

Group 6 – It works on my machine!

Name1	Name2	Name3	Name4
E0000000	E0000000	E0000000	E0000000
A0000000	A0000000	A0000000	A0000000

***Abstract*—The "Simplify News" project aims to enhance news accessibility by employing Natural Language Processing (NLP) techniques. More than one in four adults in Singapore has literacy skills at or below Level 1. This presents a considerable barrier to accessing and understanding online news, particularly for individuals with lower English proficiency. By offering condensed versions of news and tools to decipher potentially challenging vocabulary, this project aims to empower readers with more choices and facilitate deeper engagement with online information.**[1]

I. Introduction

The digital age has brought an unprecedented influx of information, with online news becoming a primary source of information. However, this accessibility is not universal. Statistics indicate that a significant portion of the population, such as more than one in four adults in Singapore, has literacy skills at or below Level 1 [1]. This presents a considerable barrier to accessing and understanding online news, particularly for individuals with lower English proficiency. Recognizing this challenge, the "Simplify News" project aims to enhance news accessibility by employing various Natural Language Processing (NLP) techniques, including simplifying language, summarizing key information, and providing visual and contextual support. This report details the progress made on two specific subtasks within this broader objective: 1) Summarizing articles by focusing on the core messages, 2) Providing support for understanding complex vocabulary. Each subtask has been approached with the selection of the most suitable dataset and methodology respectively. By offering condensed versions of news and tools to decipher potentially challenging vocabulary, this project aims to empower readers with more choices and facilitate easier and deeper engagement with online information.

Although the problem statement is framed within the Singaporean context, this project formulates the problem with the goal of general English news simplification. This foundational work will pave the way for exploring more nuanced adaptations, such as addressing the specific linguistic features of "Singlish," as a future extension of the project.

II. Task 1: Text Summarization

A. Motivation & Main Goals

This subtask addresses the central question of effectively extracting the core message from general English news articles to enhance accessibility for individuals with lower English proficiency. It focuses primarily on leveraging extractive summarization techniques, which output the most representative original sentences, to identify and present the core message of the news articles. We also explored an abstractive summarization approach using a pre-trained model to generate concise summaries beyond direct sentence extraction.

B. Data Collection and Understanding

For this subtask, we used the publicly available CNN/DailyMail dataset, accessible through the Hugging Face (dataset *abisee/cnn_dailymail*) [2] [3]. This dataset comprises a substantial corpus of English news articles from CNN and Daily Mail, paired with human-written highlights. The data is suitable for this task for several reasons:

- **General English News:** The articles provided cover a broad range of topics typical of general English news reporting, aligning with the project focus. Furthermore, this dataset provides the news text directly; utilizing this pre-existing dataset saves significant development effort for extensive web scraping and data extraction, saving significant development effort. Importantly, news articles tend to follow a structured writing style—often introducing key information early and elaborating in subsequent paragraphs—which makes them well-suited for extractive summarization tasks.
- **Scale:** This large scale dataset provides substantial data for development and evaluation. The dataset is already divided into distinct subsets: a training set comprising 287,113 articles, a validation set with 13,368 articles, and a test set containing 11,490 articles.
- **Reference Summaries**: The inclusion of human-written highlights for each article offers a valuable ground truth to evaluate the effectiveness of the extracted summaries using metrics like ROUGE or BERT Score. These reference summaries represent a benchmark for the desired output.

An initial exploration of the training dataset of CNN/DailyMail dataset reveals key characteristics regarding the length of articles and their corresponding reference summaries.

Analysis of the provided Figure 1 reveals considerable variability in the length of the news articles, ranging from very short pieces to substantially longer ones. However, a significant portion of these articles, roughly half, falls within

[1]This abstract is mainly generated by the summarization methods proposed in this project.

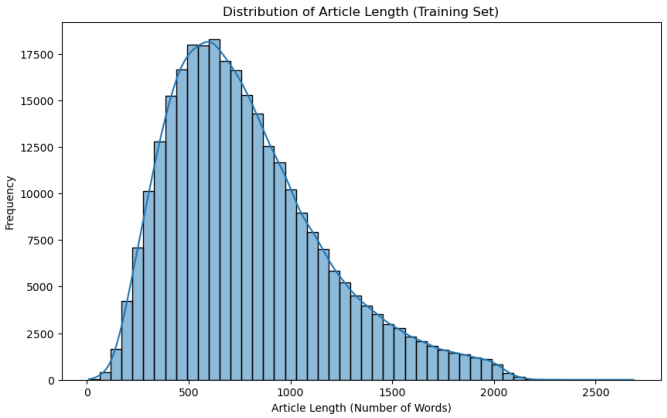

Fig. 1. Distribution of Article Length (Training Set)

the range of 500 to 1000 tokens. This wide range in article length highlights the importance of developing summarization models that can effectively handle inputs of varying sizes.

```
Summary Length Statistics (Training Set):
count    287113.000000
mean         54.786426
std          23.063941
min           4.000000
25%          40.000000
50%          51.000000
75%          64.000000
max        1974.000000
Name: summary_length, dtype: float64
```

Fig. 2. Summary Length Statistics (Training Set)

```
Summary Number of Sentences Statistics (Training Set):
count    287113.000000
mean          3.682644
std           1.353665
min           1.000000
25%           3.000000
50%           4.000000
75%           4.000000
max         107.000000
Name: summary_sentences, dtype: float64
```

Fig. 3. Summary Number of Sentences Statistics (Training Set)

In contrast, based on the statistics shown in Figures 2 and 3, the length of the summaries shows less fluctuation. A substantial portion of the summaries falls within the 40 to 60+ word count range, and the majority of the summaries are composed of approximately 3 to 4 sentences. This relatively consistent summary length and sentence count suggest that generating summaries around 4 sentences in length might be a reasonable target for our models.

C. Data Preprocessing and Preparation

To prepare the raw news dataset for extractive summarization, we first performed a series of data preprocessing steps. If we dive into the news article dataset, we can recognize the unstructured nature of news articles, characterized by varied formats, sentence counts, and the presence of extraneous characters.

- **Text Cleaning and Sentence Segmentation:** The input text is first processed to segment it into individual sentences using the *spaCy* library. Duplicate sentences are removed, and only sentences containing five or more words are retained to filter out potentially less informative segments.
- **News-Specific Formatting and Notice Removal:** In addition to general text cleaning, we implemented specific steps to handle common news article formatting and remove non-essential content. Firstly, we examined the initial ten tokens of each article for the delimiter '– ' as it often contained news agency information (e.g., "LONDON, England (Reuters)"). Secondly, we employed regular expressions to identify and remove boilerplate and legal notices frequently found in news articles. These included copyright statements, "all rights reserved" declarations, redistribution restrictions, and "e-mail to a friend" links. Finally, we also removed patterns indicative of hyperlinks embedded within the text, such as phrases starting with "Watch" and ending with a ">>" symbol.
- **Tokenization and Part-of-Speech Filtering:** Each sentence is then tokenized using spaCy. Only tokens that are alphanumeric, not stopwords, and belong to the part-of-speech categories of noun (NOUN), verb (VERB), adjective (ADJ), or proper noun (PROPN) are kept. This step aims to focus on the content-bearing words within each sentence when calculating the scores.
- **Lemmatization:** The remaining tokens in each sentence are lemmatized. Lemmatization reduces words to their base or dictionary form, helping to group together semantically related words with different surface forms. The lemmatized tokens are normalized to lowercase.

D. Summarization Methods

*1) **TextRank**:* This study investigates the application of **TextRank** as a graph-based method for extractive summarization. TextRank leverages the principles of PageRank, adapting it to rank the importance of sentences within a text based on their connections to other sentences, where connections are determined by semantic similarity.

The proposed methodology for TextRank-based summarization leverages semantic similarity derived from word embeddings and focuses on content words through part-of-speech filtering and lemmatization to identify the most important sentences for extractive summarization.

- **Sentence Similarity Calculation:** After the preprocessing steps mentioned in the above section, the semantic similarity between pairs of lemmatized sentences is calculated using the cosine similarity of their spaCy word

embeddings. Each sentence is represented by the average of the word embeddings of its constituent lemmatized tokens. This similarity score serves as the weight of the edges in the subsequent graph construction.

- **Graph-Based Ranking:** A graph is implicitly constructed where each sentence is a node. The edges between sentences are weighted by the cosine similarity calculated in the previous step. The PageRank algorithm with Random Surfer Model using *alpha=0.9* is then applied to this similarity matrix. The algorithm iteratively updates a score for each sentence based on the scores of the sentences it is similar to and the strength of those similarities.
- **Extractive Summary Generation:** The sentences are ranked based on their final TextRank scores in descending order. The top N (default to 4) ranked sentences are selected to form the extractive summary. These selected sentences are then presented in their original order of appearance in the input text to maintain coherence.

*2) **LDA**:* This project experiments with the application of **Latent Dirichlet Allocation (LDA)** as a thematic analysis method for extractive summarization. LDA is a generative probabilistic model that aims to uncover the underlying thematic structure (topics) within a collection of documents. The rationale for employing LDA lies in its ability to capture the underlying semantic structure of the text, potentially enabling the selection of sentences that are central to the core message of the article at a thematic level, beyond surface-level lexical similarities. It is hypothesized that this approach may yield more semantically coherent and informative extractive summaries.

By modeling each article as a mixture of latent topics and each word as being attributable to one of these topics, LDA can help to understand the main themes discussed within the news content. Once the topics are learned, it makes it easier to analyze how strongly each sentence in an article is associated with the most prominent topics. Sentences that exhibit a high probability of belonging to these core topics are likely to contain crucial information related to the article's main message.

- **LDA Model Application:** After the preprocessing steps, the LDA model was fitted to the preprocessed corpus using the *gensim* library in Python. A critical hyperparameter in this process is the number of latent topics to be extracted. This hyperparameter *num_topics* is dynamically determined based on the length of the input article. The rationale behind this approach is that longer articles are likely to cover a wider range of topics compared to shorter ones.
- **Sentence-Topic Distribution Analysis:** Following the fitting of the model, the topic distribution for each sentence within an article will be inferred. By treating individual sentences as short documents, their topic distribution across the learned topic space will be determined using the fitted LDA model.
- **Thematic Sentence Scoring:** The scoring mechanism is achieved by first calculating the cosine similarity between the topic distribution vectors of all pairs of sentences. Subsequently, each sentence is assigned a relevance score by summing its cosine similarity scores with all other sentences in the article. The rationale behind this scoring method is that a sentence that strongly resonates with the dominant themes of the article will exhibit higher overall similarity to the collective thematic profile of the text.
- **Extractive Summary Generation:** Finally, the sentences with the highest thematic relevance scores will be selected to constitute the extractive summary. As mentioned in the data understanding section, the number of sentences in the generated summary is set to 4.

*3) **LSA**:* **Latent Semantic Analysis (LSA)**, is an unsupervised statistic technique used to discover hidden semantic relationships between words and sentences in a document. In the task of summarization, LSA identifies the most important sentences that represent the main topics of the text by analyzing patterns of word similarities. To this point, LSA plays a similar role as TextRank, which is an extractive summarization algorithm.

Similar to TextRank, LSA transforms each sentence into a numerical representation. During preprocessing, the document is first converted into a TF-IDF matrix. Singular Value Decomposition (SVD) is then applied to factorize this matrix into three components: $A = U\Sigma V^T$, where V^T represents the topic-sentence matrix, and Σ represents the strength of each latent topic. The top-ranked topics and their corresponding sentence vectors are selected based on topic importance. N (default to 4) sentences with the highest scores are extracted from the original article and reordered according to their original positions to form the final summary.

In our implementation, we used the *sumy* library to perform the experiments. The process begins with tokenization, which segments the input text into sentences and words. We then applied the *LsaSummarizer*, which follows the LSA-based summarization pipeline and computes a relevance score for each sentence. Finally, the top n sentences with the highest scores are selected and returned as the extractive summary.

*4) **Pretrained Model**:* In order to test the performance of the pretrained models, we implemented a pretrained model **BART** [4], which is a sequence to sequence model. It is a model that combines the idea from BERT and GPT. In this project, we use a pretrained model due to limited computing power.

As a pretrained model, BART acts like a denoising autoencoder, which corrupts the initial text inputs, and reconstruct the original text from it. In the project we use the generation model with the pretrained Facebook bart-large model. We first tokenize the input to get the tensor of ids, then generate summary using tokens, and decode them into English words.

E. Evaluation

In order to have a straightforward insight on the performance of the models above, we use ROUGE and BERT Score

as the evaluation methodology.

ROUGE [5], which is Recalled-Oriented Understudy for Gisting Evaluation, is a popular metrics for evaluating text generation tasks like text summarization. It is an N-gram overlap-based metric, which means that it simply measures how much the generated summary overlaps with the reference summary in words and phrases. This means that ROUGE prefers generated summary with the same keywords and phrases as the reference summary.

BERT Score [6] is another metric which focuses on semantic similarity using contextual embeddings from BERT. It compares different tokens in the generated text with tokens in the reference summary with the similar meanings. Compared to ROUGE which only captures word similarities, BERT Score can capture semantic similarity, which recognizes paraphrasing and synonyms.

Given the methods above, we randomly test 10 articles from the dataset and test the generated summary with the human-made summary which is included in the dataset. We use the precision, recall, and F1 score to show the performance. The result is shown in Table I.

TABLE I
TASK 1 RESULT

Method	Metric	Precision	Recall	F1 Score
TextRank	ROUGE	0.504	0.277	0.354
	BERT Score	0.533	0.620	0.572
LSA	ROUGE	0.393	0.227	0.277
	BERT Score	0.543	0.598	0.568
LDA	ROUGE	0.251	0.339	0.267
	BERT Score	0.503	0.496	0.497
BART	ROUGE	0.481	0.306	0.372
	BERT Score	**0.663**	**0.684**	**0.673**

From the metric we can see that for all extractive summarization methods, the ROUGE score of Recall are very low. This means that most of the extracted sentences do not contain cetain keywords as the reference summary. However, TextRank outperforms LSA and LDA in precision, which means that the extracted summary of TextRank contains much more key information than other two methods, showing the efficiency of preprocessing. Another reason why ROUGE scores are low among all methods is that the given reference summary is very abstract, meaning that using a metric like ROUGE that only compares n-gram is not able to capture all words. So when using the BERT Score, which captures semantic meanings between words and phrases, the scores increase a lot, showing that the extracted summary indeed captures key information in the article.

For BART, which is the only abstractive summarization method, we can see that the ROUGE score is still not high, meaning that ROUGE is not able to capture the info after paraphrasing the article. When it comes to BERT Score, the semantic metrics, BART outperforms all extractive methods, showing its strong ability in paraphrasing and summarization.

F. Case Study

In order to have an intuitive comparison between models, we randomly choose an article and give the summarization given by models and given by human, which is the label. The results are shown as the figure below.[2]

Fig. 4. Summary Case

Each method demonstrates distinct characteristics:

- **TextRank** selected sentences that were factually informative and structurally coherent. It tended to favor both the beginning and ending sentences of the article. This behavior is likely influenced by the typical structure of news writing, where key information is introduced early and summarized at the end. Such sentences often exhibit stronger semantic similarity to other parts of the article, leading to higher centrality scores in the sentence similarity graph constructed by TextRank.
- **LSA** focused more on topical coverage based on word co-occurrence, but its selection sometimes lacked narrative flow. Some sentences were only weakly connected semantically, likely due to the limitations of relying solely on TF-IDF-based dimensionality reduction.
- **LDA**, though useful in identifying dominant topics, produced summaries that were less precise and occasionally incoherent. Since LDA is better at identifying themes than specific informative content, its output tended to be more general and sometimes repetitive.
- **BART**, being an abstractive model, provided the most fluent and human-like summary. It paraphrased and synthesized content rather than copying sentences directly. Its summary was shorter and more readable, though it occasionally omitted fine-grained details included in the reference summary.

G. Limitations and Future Works

A key limitation of this work is the potential inconsistency and lack of strong coherence observed in the reference summaries of the original dataset, which may have affected model training and evaluation. Furthermore, due to computational constraints, our evaluation was performed on a limited sample of the data, potentially not capturing the full performance spectrum. The text mining methods also rely on a pretrained static word embeddings, which might overlook finer semantic nuances under the specific context within documents.

[2]The image has been converted to SVG, so it remains sharp even when zoomed in.

Future research will focus on leveraging more localized Singaporean news sources like CNA and The Straits Times to better capture regional linguistic characteristics. We also aim to explore integrating contextualized word embeddings into the models for improved semantic understanding, and investigate hybrid summarization approaches combining extractive summarization and abstractive summarization. With greater computational resources, a more extensive evaluation across the full dataset will be conducted to provide a more robust performance assessment.

III. Task 2: Provide Support

A. Motivation & Main Goals

Understanding news articles relies significantly on grasping specific vocabulary and named entities like people, organizations, and locations. This presents a major challenge for many readers with diverse language backgrounds and varying levels of reading experience, including non-native speakers navigating English-language news. Research in education and reading comprehension highlights the critical role that vocabulary knowledge and familiarity with such specific references play in fully grasping written information [7] [8]. While overall text summarization addressed in the previous subtask helps reduce general complexity, it does not always resolve difficulties caused by specific difficult words or unfamiliar references encountered within the original article. Therefore, this subtask focuses on delivering targeted help directly within the original text. The aim is to offer immediate, readily available support to help readers overcome specific difficulties as they encounter them, which not only improves immediate reading comprehension and builds confidence but also provides opportunities to learn new vocabulary and knowledge in context, potentially enhancing comprehension skills over time.

To achieve this, we set the following specific technical goals:

1) Identify potentially difficult words within an article (Complex Word Identification).
2) Determine the correct meaning (sense) of these words in context and provide appropriate definitions and example sentences (Word Sense Disambiguation & Support Generation).
3) Recognize named entities within the article (Named Entity Recognition).
4) Provide further context by linking recognized named entities to relevant external resources, such as their corresponding Wikipedia page (Entity Linking).

B. Data Preprocessing

This task begins with any input news article in plain text, which then undergoes preprocessing before being passed through a series of NLP steps arranged in a pipeline. The preprocessing step mainly transforms the input string into a structured format that is suitable for the subsequent analysis pipeline. While Section IV discusses the methods used to fetch online news articles and convert them into plain text, here we focus specifically on the preprocessing steps applied once the raw text has been obtained.

We primarily used the spaCy library, chosen for its efficiency, comprehensive features, and production readiness. Notably, mainstream news articles typically follow a professional writing style with proper syntax, so standard preprocessing techniques generally produce reliable results for downstream tasks – unlike text from sources such as social media, which may require more specialized handling. The preprocessing pipeline includes the following steps:

- **Sentence Segmentation:** The input text is segmented into individual sentences, providing meaningful contextual units for later stages such as computing word embeddings within sentence contexts.
- **Tokenization:** Each sentence is split into individual tokens.
- **Part-of-Speech (POS) Tagging:** Each token is assigned a POS tag, essential for tasks like WordNet lookups, which often require POS information.
- **Lemmatization:** Tokens are lemmatized to their base dictionary forms to support dictionary lookups.
- **Additional Annotations:** Some additional attributes are retained to support later tasks, such as whether a token is part of a named entity, whether it contains only alphabetic characters, its character offset in the original text, etc. For example, only alphabetic tokens will be considered in complexity analysis.

The output of this preprocessing stage is a structured representation of the article text – organized as a list of sentences, each containing a list of tokens with their original form and all relevant linguistic annotations.

C. Pipeline

*1) **Complex Word Identification**:* Complex Word Identification (CWI) aims to automatically detect words within a text that may pose comprehension challenges for the intended audience. It is one of the first and most essential steps in this simplification task's pipeline. Framed as a classification task, CWI has traditionally relied on heuristic or threshold-based methods. These typically include using word frequency lists (assuming less frequent words are more complex), checking word length or syllable counts, or verifying if a word is absent from a predefined list of simple vocabulary. While straightforward, these methods typically disregard word context. More recent approaches have leveraged machine learning and deep learning techniques.

In this task, we first adopted the CWI model presented by Gooding and Kochmar [9], which treats CWI as a sequence labeling task [10]. Their method uses a Bidirectional Long Short-Term Memory (BiLSTM) network architecture to generate contextualized vector representations for each token. These representations are then passed through a final classification layer to predict the probability of each word being complex. We chose this model primarily because it is ready to use without the need for retraining and was developed and validated on the dataset by Yimam et al. [11], which

includes texts from news sources – closely aligning with the type of content targeted in our project.

In addition to a model-based approach, we also employed a more traditional and straightforward method to identify complex words based on predefined vocabulary lists. We noticed that people who learned English as a second language (especially those educated through local education systems) often build their core vocabulary mainly via textbooks and standardized tests. This implies that their perception of difficult words may differ from that of native speakers, who naturally pick up vocabulary from movies, social media, and daily conversations. Therefore, instead of solely relying on complex classification models, the alternative system uses a custom vocabulary list compiled from China's national education standards, covering words required from junior high school up to graduate-level exams. Within this framework, words appearing in the curriculum lists up to a certain level are labeled as "basic," while those outside are marked as potentially "hard."

This method offers several advantages. First, though initially utilizing China's educational data, it can be easily extended by integrating localized academic vocabulary lists from other countries' curricula. Second, it allows the vocabulary difficulty threshold to be dynamically tailored according to a user's specified education level (e.g., primary school terms for children versus university-level lexicon for undergraduates). Third, using curriculum standards provides a more standardized way to gauge complexity based on learners' expected vocabulary, rather than relying solely on subjective difficulty judgments.

*2) **Word Sense Disambiguation & Support Generation**:* Word Sense Disambiguation (WSD) has been a focus of relevant researchers for a long time [12], and many approaches have been proposed to solve this problem. In general, there are mainly 3 types of approaches. The first one is the Knowledge-Based approach (KB) [13]. The method relies on pre-built semantic resources like WordNet to disambiguate word senses by comparing contextual patterns with definitions, example sentences, and semantic relationships in the knowledge base. The second one is the Vector-Based approach. This method begins with encoding words into embeddings, then uses other supervised machine learning techniques like KNN, classifiers, etc. [14], to find the most suitable senses. The third one is also a supervised technique, but instead of disambiguating with word embeddings, it trains the model on the relationship within a graph knowledge database. The method then performs a procedure on the knowledge graph database and gets the final result (e.g., PageRank) [16].

In our implementation, we adopted a simple method that is similar to an approach mentioned by another research paper [14]. The method first gets the target word's BERT [18] embedding within the context sentence. Then, for each possible sense with the same POS tag, it combines a new sentence "target word, sense definition." and calculates the word embedding for each sense. Finally, it compares the embedding generated by each sense with the original embedding and picks the sense with the largest cosine similarity. The pseudo-code is shown in Algorithm 1.

Algorithm 1 WSD Algorithm

Input: target word w, context sentence S, pos tag POS, dictionary $\mathcal{D}(w, \text{POS}) = \{s_1, s_2, ..., s_k\}$

Output: Best sense s^*

Initialize pretrained BERT model $\mathcal{M}_{\text{BERT}}$
Feed S into $\mathcal{M}_{\text{BERT}}$, obtain contextual embeddings $H = [h_1, h_2, ..., h_n]$
Locate position index i of w in S, extract embedding $h_w = H[i]$
for each sense $s_j \in \mathcal{D}(w, \text{POS})$ **do**
    Construct definition sentence $S_j^{\text{def}} =$ "w, s_j."
    Feed S_j^{def} into $\mathcal{M}_{\text{BERT}}$, obtain w token embedding h_j^{def}
    Compute cosine similarity $\text{sim}_j = \frac{h_w \cdot h_j^{\text{def}}}{\|h_w\| \|h_j^{\text{def}}\|}$
    Store similarity score $\text{sim}_{all} \leftarrow \text{sim}_j$
end for
Select best sense $s^* = \arg\max_{s_j}(\text{sim}_{all})$ **return** s^*

Once the most probable sense is identified using the WSD algorithm, its corresponding WordNet definition can be retrieved. However, it is noticed that WordNet definitions can sometimes be technical to understand. To explore alternative forms of support, we also experimented with generating contextualized, natural language definitions using a large language model. Specifically, we utilized a Flan-T5 model fine-tuned for definition generation, based on the work of Giulianelli et al. [15] and available via Hugging Face (model *ltg/flan-t5-definition-en-large*). Following the approach described in the paper, this model accepts a prompt containing the original sentence where the complex word appears, appended with the question "What is the definition of [complex word]?". The model then generates a textual definition sequence tailored to the word's usage in the specific sentence.

*3) **Name Entity Recognition and Linking**:* For Named Entity Recognition (NER), current methods primarily fall into traditional approaches and deep learning approaches. Traditional methods include rule-based systems [17], which rely on manually designed patterns but lack generalization, and statistical models [19], which learn entity patterns from annotated data. Deep learning approaches like BERT [20] leverage neural networks to automatically capture contextual features, achieving higher accuracy without manual rules, though they demand significant computational resources and labeled data. In our implementation, we simply use spaCy's NER mechanism. It relies on pretrained deep learning models, primarily using CNNs with word embeddings and subword features for entity detection.

For Entity Linking, we employed a third-party package *spacy-entity-linker* [21]. This package is an extension to the original spaCy pipeline and uses the Wikidata database to identify all entities and link them to all possible Wikidata URLs, which are disambiguated by embedding similarity calculations.

Combining the above procedures into one, we generate the NER entity with spaCy's original identifier and perform entity

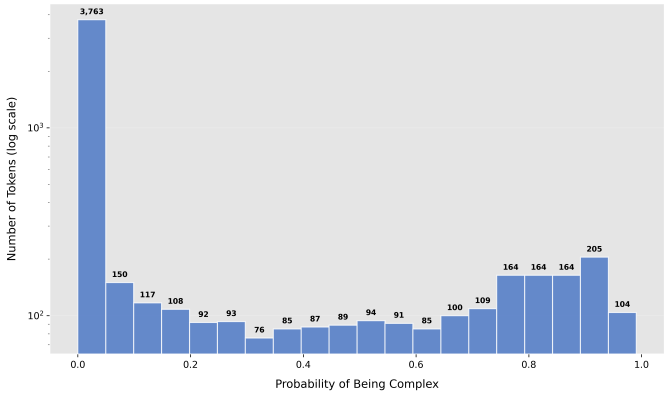

Fig. 5. Distribution of predicted complexity probabilities for all tokens in five sample news articles using the BiLSTM sequence labeling model. Probabilities below 0.05 typically correspond to stopwords.

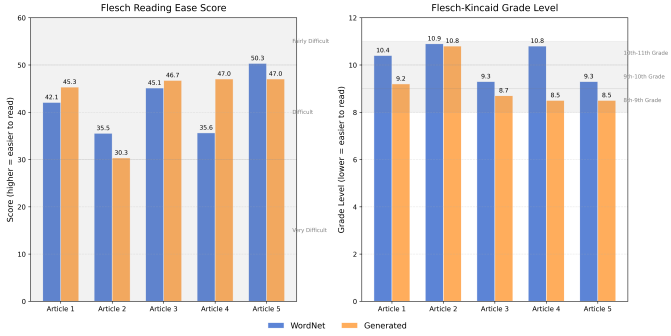

Fig. 6. Comparison of readability metrics between WordNet and Flan-T5-generated definitions

linking with spacy-entity-linker respectively. Then, we check the overlapping tokens they identify and return this as the final result of NER identification.

D. Evaluation and Discussion

Evaluating the effectiveness of the "Provide Support" pipeline is inherently challenging, given the application-specific nature of this task. Since the task is largely unsupervised, starting from real articles rather than labeled datasets, some of the assessments in this section is qualitative and intended to offer insights based on outputs from several example articles. Ideally, user studies involving real readers would provide more robust feedback on the actual helpfulness and accuracy of the provided support mechanisms.

Firstly, some limitations were observed with the CWI model used in the pipeline, adapted from Gooding and Kochmar [9]. In their original work, a fixed probability threshold of 0.5 was used for binary classification of word complexity. However, our experiments with sample news articles suggest that this threshold could yield high recall but relatively low precision. As illustrated in Figure 5, applying a 0.5 threshold results in approximately 60% of the remaining words (excluding low-probability stopwords) being labeled as complex. While prioritizing recall helps ensure genuinely difficult words are not missed, excessive marking may distract readers and reduce overall usability. Additionally, the model's raw probability outputs do not appear to reliably reflect the true gradation of word complexity, limiting their use as a continuous scoring signal.

Secondly, the current pipeline's capability in handling multi-word phrases is limited. A primary issue lies in reliably identifying relevant multi-word expressions (such as noun or verb phrases) suitable for complexity assessment, beyond the basic compound words detected using spaCy's parser. Furthermore, even after identifying potential multi-word phrases, effectively assessing their complexity is challenging. Although the original CWI paper mentions predicting phrase complexity by combining the scores of individual words, it does not specify the exact implementation. In practice, such aggregation can be unreliable by failing to capture the nuanced complexities of phrases. Addressing these limitations may require incorporating comprehensive graded phrase dictionaries or developing machine learning models specifically for phrase extraction and complexity prediction (potentially utilizing phrasal embeddings).

Thirdly, the quality of definitions provided for difficult words remains a major area for improvement. Definitions sourced from WordNet are often overly verbose or technical, frequently introducing additional complex terms that reduce their effectiveness in a simplification context. In contrast, definitions generated by the Flan-T5 model tend to be more readable but occasionally suffer from accuracy issues, such as including the target word itself or failing to provide meaningful clarification. As shown in Figure 6, Flan-T5-generated definitions achieve better Flesch-Kincaid readability test scores [22], as they are typically shorter and composed of simpler vocabulary. One promising direction for improving definition quality and accessibility is to replace WordNet with alternative sources, such as child-friendly or elementary-level dictionaries, as input to the proposed WSD algorithm. Notably, since the algorithm operates on word embeddings and is agnostic to the source of definitions, integrating simpler explanations via external APIs is both feasible and effective. Nonetheless, effectively evaluating the impact of the provided definitions on overall article readability remains a non-trivial task. Standard readability metrics are sensitive to factors like sentence length, which leads to skewed results when using approaches like inline definitions that greatly increase sentence length.

IV. Visualization

To present the results of our news simplification process effectively, we developed a simple frontend webpage[3]. This interface provides a direct view of the results of both subtasks, as shown in Figures 7, 8, and 9. After input text is processed,

[3]The code for the visualization system is available at https://github.com/peter-233/cs5246-project.

the visualization highlights difficult words with red bars and named entities with orange bars. Users can hover over the highlighted terms to access additional information: difficult words display their definition and an example sentence, while named entities show their label and a link to the Wikipedia page. In addition, users can select a summarization method from TextRank, LDA, LSA, and Bart to generate a concise summary of the input text.

To support this functionality, the system must first acquire the text of the target article. We support two main input methods. Firstly, users can provide the article content directly as plain text. Secondly, to enable real-world application and testing, we implemented functionality to accept a news article URL.

Currently, the URL-handling capability was specifically developed and tested using Channel NewsAsia (CNA) articles. When a URL is provided, the system fetches the raw HTML and uses the *BeautifulSoup* library to parse it. The extraction targets the title, description, and main body, using manually identified CSS selectors that reflect the common structure of CNA articles. Basic filtering is also applied to remove non-content text. It is worth noting that this web extraction component is both standalone and site-specific, so it can be extended to support additional news sources by adapting the parsing logic. In contrast, all subsequent analysis pipelines are designed to operate independently on the resulting text, whether it is input directly or extracted from a URL.

Fig. 7. Difficult Word Support Visualization

Fig. 8. NER Visualization

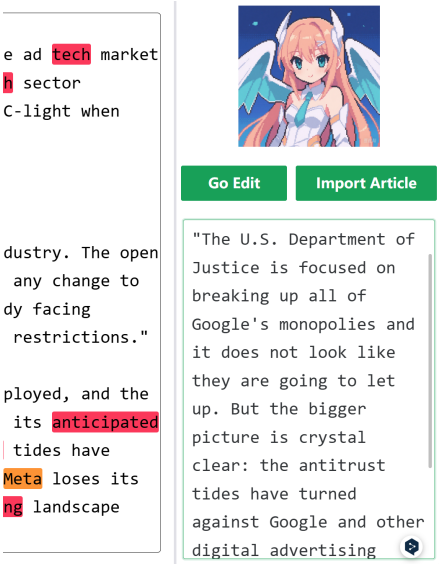

Fig. 9. Summary Visualization

V. Conclusion

This project presents a practical approach to addressing the challenge of news accessibility for readers with lower English proficiency by combining two complementary sub-tasks: text summarization and contextual support provision. The summarization component explores various extractive and abstractive techniques, including traditional algorithms like LDA, TextRank, and LSA, as well as a powerful pretrained model, BART. The support component introduces a robust pipeline that identifies complex words and named entities, disambiguates word senses using context-aware methods, and provides reader-friendly explanations and external references. Furthermore, the frontend user interface of the system allows users to interactively access simplification features, bridging the gap between technical NLP tools and real-world usability. While our current implementation delivers a functional and user-friendly system, both news summarization and contextual support remain complex NLP challenges that warrant further exploration to enhance its effectiveness and impact.

REFERENCES

[1] "Survey of Adults Skills 2023: Singapore," *OECD*, 2024. https://www.oecd.org/en/publications/survey-of-adults-skills-2023-country-notes_ab4f6b8c-en/singapore_382e963a-en.html

[2] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2017, pp. 1073–1083. doi: 10.18653/v1/P17-1099.

[3] K. M. Hermann *et al.*, "Teaching Machines to Read and Comprehend," in *NIPS*, 2015, pp. 1693–1701. [Online]. Available: http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend

[4] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ArXiv Preprint ArXiv:1910.13461. https://huggingface.co/facebook/bart-large-cnn

[5] Lin, C. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*. pp. 74-81 (2004)

[6] Zhang, T., Kishore, V., Wu, F., Weinberger, K. & Artzi, Y. Bertscore: Evaluating text generation with bert. *ArXiv Preprint ArXiv:1904.09675*. (2019)

[7] G. Brooks, J. Clenton, and S. Fraser, "Exploring the importance of vocabulary for English as an additional language learners' reading comprehension," *Studies in Second Language Learning and Teaching*, vol. 11, no. 3, pp. 351–376, Sep. 2021, doi: 10.14746/ssllt.2021.11.3.3.

[8] R. Smith, P. Snow, T. Serry, and L. Hammond, "The Role of Background Knowledge in Reading Comprehension: a Critical Review," *Reading Psychology*, vol. 42, no. 3, pp. 214–240, 2021, doi: 10.1080/02702711.2021.1888348.

[9] S. Gooding and E. Kochmar, "Complex Word Identification as a Sequence Labelling Task," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Jul. 2019, pp. 1148–1153. doi: 10.18653/v1/P19-1109.

[10] M. Rei, "Semi-supervised Multitask Learning for Sequence Labeling," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2017, pp. 2121–2130. doi: 10.18653/v1/P17-1194.

[11] S. M. Yimam, S. Štajner, M. Riedl, and C. Biemann, "CWIG3G2 - Complex Word Identification Task across Three Text Genres and Two User Groups," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Nov. 2017, pp. 401–407. [Online]. Available: https://aclanthology.org/I17-2068/

[12] Weaver, W. 'Translation'. In: Locke, W.N. and Booth, A.D. (eds.) *Machine Translation of Languages: Fourteen Essays*. Cambridge, Mass.: MIT Technology Press, 1955, pp. 15-23.

[13] Agirre, E., Lopez de Lacalle, O., and Soroa, A. *Random Walks for Knowledge-Based Word Sense Disambiguation*. Computational Linguistics, 2014, pp. 57–84.

[14] Peters, M.E., Neumann, M., Iyyer, M., et al. *Deep Contextualized Word Representations*. In: Proceedings of NAACL-HLT 2018. New Orleans: ACL, 2018, pp. 2227–2237.

[15] M. Giulianelli, I. Luden, R. Fernandez, and A. Kutuzov, "Interpretable Word Sense Representations via Definition Generation: The Case of Semantic Change Analysis," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2023, pp. 3130–3148. doi: 10.18653/v1/2023.acl-long.176.

[16] Scozzafava F., Maru M., Brignone F., et al. *Personalized PageRank with Syntagmatic Information for Multilingual Word Sense Disambiguation*. In: ACL 2020 System Demonstrations. Online: ACL, 2020, pp. 135–141.

[17] Collins, M., & Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (pp. 100-110). EMNLP.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[19] Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning* (pp. 282-289). ICML.

[20] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics* (Vol. 1, pp. 4171-4186). NAACL.

[21] Egerber. *spaCy-entity-linker (v1.0.0)* [Software]. GitHub, 2023. URL: https://github.com/egerber/spaCy-entity-linker

[22] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel," National Technical Information Service, Springfield, Virginia 22151 (AD-A006 655/5GA, MF $2.25, PC $3.75), Feb. 1975. [Online]. Available: https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=1055&context=istlibrary

APPENDIX

All team members believe that each person contributed equally to the project. A detailed breakdown of individual contributions is provided in Table II.

TABLE II
MEMBERS CONTRIBUTION

Member	Contribution
Name1	Provide Support Task
Name2	Text Summarization Task
Name3	Text Summarization Task
Name4	Provide Support Task