Segment First or Comprehend First? Explore the Limit of Unsupervised Word Segmentation with Large Language Models

Zihong Zhang[1], Liqi He[2], Zuchao Li[1,*], Lefei Zhang[2], Hai Zhao[3], Bo Du[2]

[1]School of Artificial Intelligence, Wuhan University, Wuhan, China
[2]School of Computer Science, Wuhan University, Wuhan, China
[3]School of Computer Science, Shanghai Jiao Tong University, China
{zhangzihong, heliqi, zcli-charlie, zhanglefei, dubo}@whu.edu.cn
zhaohai@cs.sjtu.edu.cn

Abstract

Word segmentation stands as a cornerstone of Natural Language Processing (NLP). Based on the concept of "comprehend first, segment later", we propose a new framework to explore the limit of unsupervised word segmentation with Large Language Models (LLMs) and evaluate the semantic understanding capabilities of LLMs based on word segmentation. We employ current mainstream LLMs to perform word segmentation across multiple languages to assess LLMs' "comprehension". Our findings reveal that LLMs are capable of following simple prompts to segment raw text into words. There is a trend suggesting that models with more parameters tend to perform better on multiple languages. Additionally, we introduce a novel unsupervised method, termed LLACA (**L**arge **L**anguage Model-Inspired **A**ho-**C**orasick **A**utomaton). Leveraging the advanced pattern recognition capabilities of Aho-Corasick automata, LLACA innovatively combines these with the deep insights of well-pretrained LLMs. This approach not only enables the construction of a dynamic n-gram model that adjusts based on contextual information but also integrates the nuanced understanding of LLMs, offering significant improvements over traditional methods. Our source code is available at https://github.com/hkr04/LLACA

1 Introduction

Understanding language is the core task of NLP (Zhang et al., 2019). To measure the understanding capabilities of language models, various Natural Language Understanding (NLU) tasks have been proposed, such as Question Answering (Namazifar et al., 2021; Yao et al., 2023; He et al., 2024) and Sentiment Classification (Arevalillo-Herráez et al., 2022; Jing et al., 2021). However, most of these tasks only assess the language model's understanding of the overall meaning of sentences, lacking an evaluation of the language model's understanding capabilities at a fine-grained level.

In many languages, such as Chinese and Japanese, there are no explicit word boundaries. Therefore, word segmentation is a crucial foundational step in NLP tasks like syntactic analysis (Cereda et al., 2018; Li et al., 2018), information retrieval (Ponte and Croft, 2017), and machine translation (Moslem, 2024; Li et al., 2020) for these languages. Most previous research on word segment has adhered to the principle of "segment first, comprehend later", because word segment has long been regarded as the first step in NLP (Zhao et al., 2006). However, the human brain's process of analyzing sentences typically involves an interactive process of segmentation and comprehension, where segmentation depends on comprehension and vice versa, especially in sentences with ambiguity. Therefore, word segmentation can also be the last step in NLP, that is, to test the understanding capabilities of a language model.

Large Language Models (LLMs) are the best linguists. The emergence of LLMs has marked significant advancements in NLU. And the capabilities of LLMs are no longer based on word segmentation. However, word segmentation can serve as an indicator to effectively evaluate the semantic understanding capabilities of LLMs. Additionally, we can utilize LLMs for word segmentation, and we propose a word segment framework named LLM-Word Segmentation (LLM-WS). We employ the framework to explore the limit of unsupervised word segmentation with LLMs and evaluate the semantic understanding capabilities of LLMs based on word segmentation.

Our research demonstrates that LLMs can per-

* Corresponding author. This work was supported by the National Natural Science Foundation of China (No. 62306216), the Natural Science Foundation of Hubei Province of China (No. 2023AFB816) and the Fundamental Research Funds for the Central Universities (No. 2042025kf0026).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18146–18163

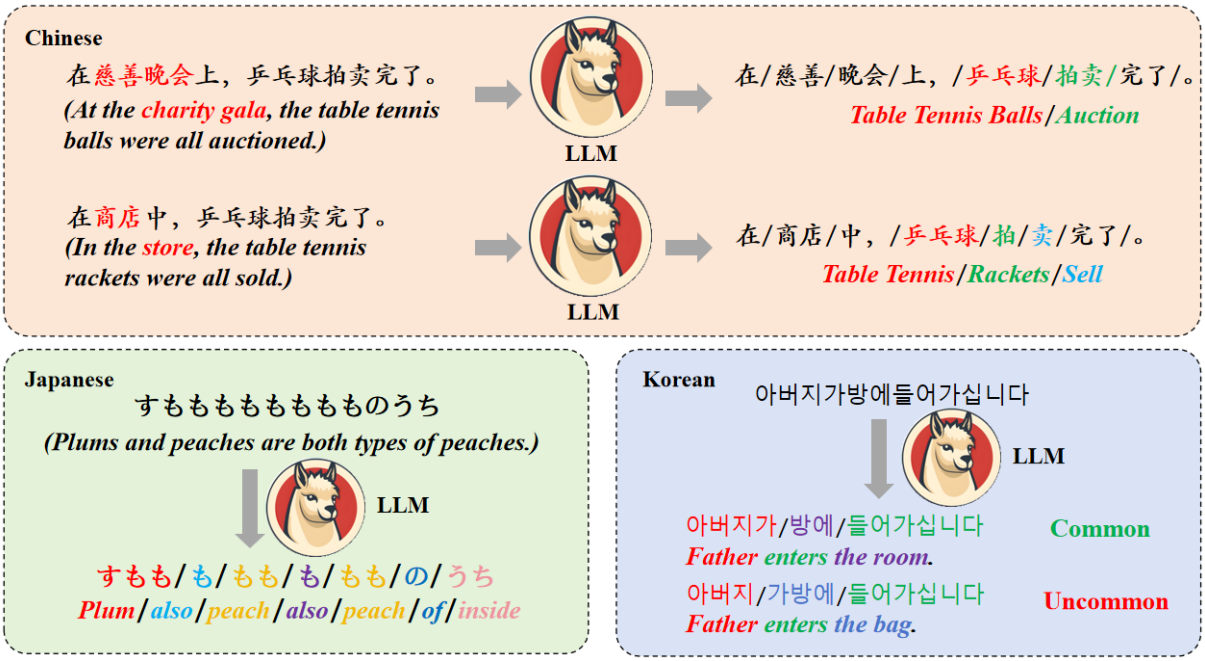

Figure 1: Examples of LLM-WS

form word segmentation on raw text based on simple prompts, as shown in Figure 1. In the upper section of Figure 1, we demonstrate how LLMs can accurately segment typical Chinese sentences with ambiguity by taking context into account. Based on the different scenarios of "charity gala" and "store", the two identical Chinese word sequences "ping pong balls are auctioned" and "ping pong racket is sold" were correctly segmented by LLMs. The example in the bottom left corner of Figure 1 illustrates that LLMs can correctly segment consecutive identical characters within a sentence. The example in the bottom right corner of Figure 1 illustrates that LLMs can segment ambiguous sentences by considering the likelihood of different interpretations within the actual context. Undoubtedly, "father enters the room" is more common than "father enters the bag". The aforementioned examples also demonstrate that LLMs possess capabilities of word segmentation across multiple languages, including Chinese, Japanese, and Korean.

In previous word segmentation research, supervised and unsupervised methods are two main learning paradigms for word segmentation. While supervised methods have indeed demonstrated remarkable efficacy (Zhao et al., 2019), they encounter hurdles such as a heavy reliance on a great deal of manually labeled corpora and poor domain adaptability. Unsupervised word segmentation approaches can avoid the need for a large amount of human labor required for labeled datasets. Additionally, previous research has demonstrated that unsupervised word segmentation methods have better stability for unseen words and adaptability to new domains. Previous unsupervised methods can be broadly classified into two types: discriminative models and generative models. The former evaluates the quality of word candidates using carefully designed goodness measures. However, this method lacks the capability to handle ambiguous strings, which is a major source of segmentation errors. The latter designs models to find the optimal segmentation with the highest generative probability. These methods have better stability in segmenting ambiguous sentences. With the rapid development of neural networks in recent years, research has been conducted to perform unsupervised word segmentation using neural generative models (Wang et al., 2017; Sun and Deng, 2018), achieving competitive performance to the state-of-the-art statistical models. The most significant difference between our approach and previous research is the implementation of comprehension-based word segmentation. This represents a new era in the development of word segmentation methods. Based on LLMs trained on massive corpora, our framework named LLM-WS can explore the limit of unsupervised word segmentation.

Specifically, in LLM-WS, we propose an unsupervised word segmentation method named

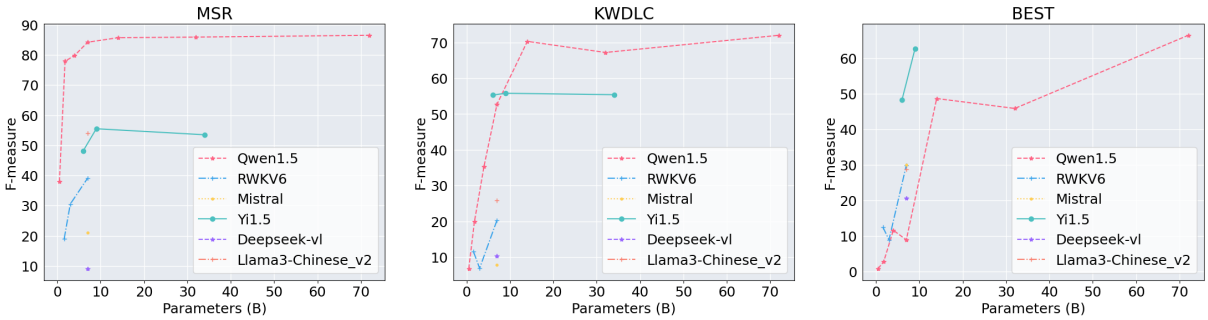

Figure 2: Evaluations on datasets MSR (Simplified Chinese), KWDLC (Japanese) and BEST (Thai).

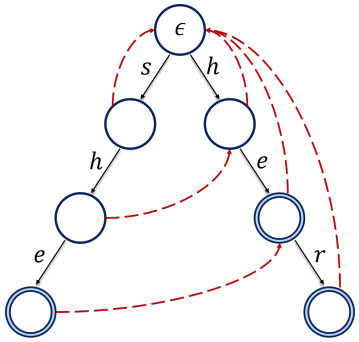

Figure 3: The illustration of an Aho-Corasick automaton with patterns "she", "he", and "her", with final states in double circles and failure links in red.

LLACA (**L**arge **L**anguage Model-Inspired **A**ho-**C**orasick **A**utomaton). Leveraging the profound insights into language provided by well-pretrained LLMs, we utilize the advanced pattern recognition capabilities of Aho-Corasick automata (Aho and Corasick, 1975) to achieve unsupervised word segmentation. We utilize LLMs to conduct word segmentation within the corpus and compute word frequencies. Subsequently, we integrate the computational outcomes derived from LLMs with the AC automaton to facilitate efficient unsupervised word segmentation.

Our findings reveal that larger language models often exhibit stronger segmentation capabilities. Conversely, models with fewer parameters frequently struggle to adhere to segmentation prompts, leading to significant hallucinations and deviations from human-annotated gold standards. Notably, the Chinese LLM Qwen1.5-7B-Chat has already surpassed previous state-of-the-art results on the Chinese word segmentation tasks for the MSR and PKU datasets.

2 Related Work

For word segmentation, the simplest but effective method is the maximum matching model (Jurafsky and Martin, 2014). Beginning at the start of a string, the maximum matching model selects the longest dictionary word that corresponds to the current position and then moves forward to the end of that matched word within the string. However, it is clear that this method cannot recognize words that are not included in the dictionary (Huang and Zhao, 2007).

With the rise of statistical machine learning methods, word segmentation is formalized as a sequence labeling task. Traditional sequence labeling models such as Hidden Markov Models (HMM) (Carpenter, 2006; Yan et al., 2021), Maximum Entropy Markov Models (MEMM) (McCallum et al., 2000) and Conditional Random Fields (CRF) (Lafferty et al., 2001; Peng et al., 2004; Zhao et al., 2006) are widely used. The CRF becomes the mainstream method of supervised word segmentation. Multiple variants of CRF formed the standard word segment models before the deep learning era. The linear-chain CRF model is based on the Markov assumption, where the current state depends only on the previous state and the observation sequence, which is not conducive to word segmentation on longer sequences. The first implementation of semi-CRF for word segmentation was published in 2006 (Andrew, 2006). Semi-CRF with latent variables was applied to word segmentation, significantly enhancing the performance of CRFs (Sun et al., 2009, 2012). CRFs can achieve higher word segmentation accuracy, but the complexity of the model leads to slightly lower segmentation efficiency.

Most of unsupervised word segmentation methods can be categorized into two types: goodness-based methods and nonparametric Bayesian meth-

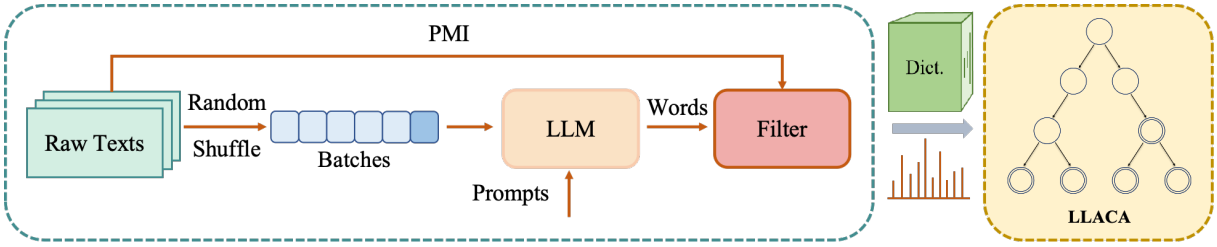

Figure 4: The construction of LLACA.

ods. ESA is a goodness-based method for unsupervised Chinese word segmentation, employing an iterative process with local maximum strategy (Wang et al., 2011). nVBE relies on Variation of Branching Entropy, enhancing performance through normalization and Viterbi decoding while simplifying the model by reducing parameters and thresholds (Magistry and Sagot, 2012). One of the disadvantages of goodness measure based methods is that theoretically, they lack the ability to resolve ambiguity. Nonparametric Bayesian methods, such as those proposed by Goldwater et al. (Goldwater et al., 2009), a unigram and bigram model based on Dirichlet process and hierarchical Dirichlet process (Teh et al., 2004). The primary limitation is that the Gibbs sampler requires nearly 20,000 iterations to achieve convergence. Inspired by the "products of experts" idea, a joint model for unsupervised word segmentation combines word-based hierarchical Dirichlet process model with character-based hidden Markov model. The method achieves an unsupervised word segmentation approach that effectively solves ambiguity by calculating the product of probabilities of two generative models and employing Gibbs sampling during the inference process (Chen et al., 2014).

In recent years, deep neural networks have achieved success in a variety of tasks. Applying methods such as Recurrent Neural Networks (RNNs) (Chen et al., 2015; Sun and Deng, 2018) and Long Short-Term Memory Networks (LSTMs) (Cai and Zhao, 2016; Yao and Huang, 2016; Wang and Zheng, 2022) to word segmentation can better utilize context and reduce the extensive manual work required for feature engineering. However, neural word segmenters not only require a large amount of training corpora, but also entail more time costs for both training and inference.

3 Approach

To assess the word segmentation capabilities of LLMs, we conducted preliminary experiments using popular open-source LLMs on three datasets from languages without clear word boundaries: MSR (Emerson, 2005) for Simplified Chinese, KWDLC (Hangyo et al., 2012) for Japanese and BEST (Kosawat et al., 2009) for Thai. Accuracies are quantified using the token F-measure, which is formulated as follows. The F-measure, F, is a harmonic mean of precision and recall, where precision (P) is the ratio of correctly identified words to the total number of words identified in the output, and recall (R) is the ratio of correctly identified words to the total number of words in the gold standard.

The experiments involved simple prompts directing the LLMs to segment texts into words using spaces without further explanation or contextual information.

3.1 Preliminary Experiments on LLM-WS

The results indicate that models with fewer parameters performed more poorly. Examination of their outputs revealed two primary issues: firstly, these models completely failed to comprehend the prompts, resulting in chaotic outputs that included repeated or incorrectly replaced words; secondly, although some models partially understood the prompts and attempted text segmentation, various misalignments occurred, such as inappropriate translations and misuse of delimiters.

LLMs with more parameters, like 7 billion, generally demonstrated an adequate understanding of the task when processing texts in languages consistent with their training corpus. As the parameter count increased, hallucinations diminished, and F scores improved across different languages.

3.2 LLACA: Large Language Model-Inspired Aho-Corasick Automaton

The Aho-Corasick automaton (AC automaton), as defined by Aho and Corasick (1975), extends the concept of a Trie, a tree-like data structure commonly used for storing strings where each node

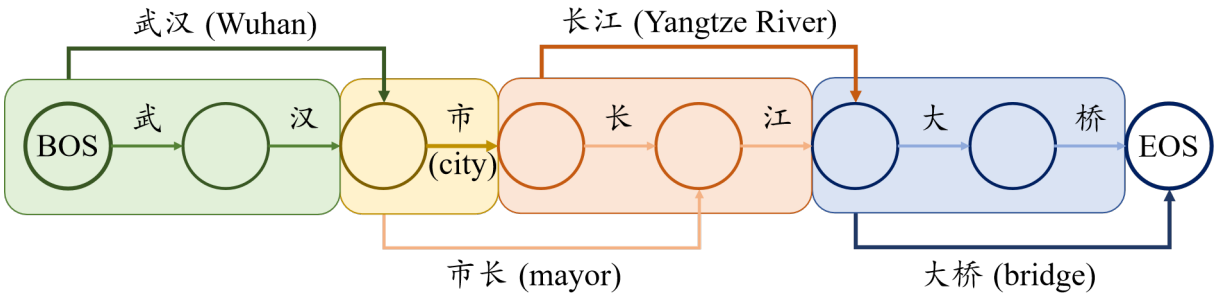

Figure 5: A Directed Acyclic Graph (DAG) for sequence segmentation, starting from "BOS" (Beginning of Sentence) and ending at "EOS" (End of Sentence). Edges denote recognized patterns. Paths through the graph represent potential segmentations, with the preferred path highlighted in frames.

signifies a string prefix, and each directed edge represents an input character or byte. Patterns are incorporated into the Trie, with each path from the root to a node denoting a pattern prefix. Notably, if a node's path represents an entire pattern, it is designated as a final state. Each node in the Trie has added failure links directing the automaton to the next node to consider if a character does not match at the current node, effectively allowing the automaton to skip unnecessary nodes and thus speeding up the pattern-matching process. Figure 3 illustrates a typical AC automaton configured with patterns "she", "he", and "her". Both the construction and recognition processes of the automaton operate with linear time complexity relative to the length of the input text.

Traditionally, methods like maximum length matching and maximum probability matching for word segmentation have leveraged AC automata in conjunction with human-annotated dictionaries. However, the dynamic nature of language and the considerable effort required for human annotations present substantial challenges.

To address these issues, our approach integrates the deep comprehension capabilities of LLMs to develop vocabularies that requires no human annotation. By harnessing the understanding inherent in LLMs, we can dynamically adapt to language changes over time, significantly reducing the labor and limitations associated with manual methods. This fusion not only enhances the automaton's application scope but also paves the way for more autonomous and robust text processing tools.

The initial step involves sampling from a LLM. We begin by randomly shuffling the raw sentences into several batches, each approximately the size of the square root of the total number of sentences. For each batch, we use simple prompts to direct the LLM to perform LLM-WS. This process may occasionally lead to errors, such as the LLM mistaking the task for translating into its primary training language, commonly referred to as "hallucinations". To address this, a filtering step is crucial. We employ Pointwise Mutual Information (PMI) to assess the coherence of each "word" $w[1..n]$ identified by the LLM. In Equation 1, $p(\cdot)$ is computed as a unigram probability.

$$\mathrm{PMI}(w) = \min_{i=1}^{n-1} \left(\log \frac{p(w[1..n])}{p(w[1..i]) \cdot p(w[i+1..n])} \right) \quad (1)$$

An important hyperparameter in this context is the "top ratio", which determines the proportion of words to retain. To minimize the risk of missing out-of-vocabulary (OOV) items, this parameter should not be set too low.

Figure 4 illustrates the construction of LLACA. Once constructed, it analyzes the semantic patterns offered by the LLM-WS. Unlike the widely-used open-sourced Chinese word segmentation system Jieba[1], which models the probability of a word occurring as $p(w) = \frac{\mathrm{count}(w)}{\sum \mathrm{count}(w_i)}$, we adopt a different approach. We define the previous state $\mathrm{prev}(w)$ as the closest final state on the path from the root to the current state w. For example, as shown in Figure 3, the previous state of "her" is "he". And the root, representing ϵ, is considered a special final state. Thus the previous state of "she" is ϵ. The denominator is now defined as the sum of word counts in the sub-Trie of $\mathrm{prev}(w)$. In other words, the probability of w is affected by its occurrence count and those of others with the same prefix $\mathrm{prev}(w)$.

$$p(w) = \frac{\mathrm{count}(w)}{\sum_{\mathrm{prev}(w) \in \mathrm{prefix}(w_i)} \mathrm{count}(w_i)} \quad (2)$$

[1] https://github.com/fxsjy/jieba

Table 1: F-measure on multilingual datasets compared with previous state-of-the-art models. "Uni" refers to the integration of LLM-WS and the unigram probability model, while "LLACA" denotes our newly proposed method. The asterisk (*) indicates evaluations performed after conversion from Traditional to Simplified Chinese using OpenCC. And the dagger (†) means the results are not directly comparable due to variations in the processing of the dataset. LLMs listed here are all Chat version.

Method		Chinese				Japanese		Korean	Thai	
		AS	**CITYU**	**MSR**	**PKU**	**KWDLC**	**UD_JA**	**UD_KO**	**BEST**	**UD_TH**
Baselines	nVBE (2012)	76.6	76.7	81.3	80.0	-	-	-	-	–
	NPY-2 (2009)	-	82.4	80.2	-	-	-	-	-	-
	NPY-3	-	81.7	80.7	-	-	-	-	-	-
	Joint (2014)	-	-	81.7	81.1	-	-	-	-	-
	SLM-2 (2018)	79.4	78.2	78.5	80.2	-	-	-	-	-
	SLM-3	80.3	80.5	79.4	79.8	-	-	-	-	-
	SGB-A-12 (2022)	-	-	-	-	-	-	-	80.1†	-
	SGB-C-4	81.0	80.0	74.0	80.0	-	-	-	-	-
	SGB-C-5	**82.4**	78.5	80.4	78.4	-	-	-	-	-
	PYHSMM (2015)	-	**82.6**	**82.9**	**81.6**	-	-	-	**82.1**†	-
Qwen1.5-7B	LLM	78.4*	82.5*	84.2	86.7	52.7	49.8	36.8	8.9	21.3
	Uni	84.6*	85.9*	86.4	87.5	64.9	58.8	43.2	37.5	32.7
	LLACA	**84.8***	**86.5***	**86.7**	**87.7**	**66.4**	**62.8**	**48.3**	**48.6**	**39.2**
Qwen1.5-14B	LLM	78.0	72.1	85.7	87.0	70.3	64.6	43.9	48.7	57.1
	LLACA	**86.6**	**82.7**	**87.8**	**89.3**	**76.7**	**69.2**	**51.9**	**64.7**	**62.2**
Qwen1.5-32B	LLM	80.5	74.9	85.9	86.3	67.2	64.3	48.3	45.9	53.8
	LLACA	**86.8**	**84.0**	**87.6**	**87.6**	**74.1**	**68.0**	**54.9**	**64.9**	**61.4**
Qwen1.5-72B	LLACA	**88.3**	**88.1**	**88.2**	**88.7**	**76.1**	**69.4**	**58.9**	**68.9**	**70.0**

When no other pattern serves as a prefix for a given pattern, the model reverts to using unigram probabilities. When the longest prefix of a word w, such as "he" for "her", reaches a final state, the model effectively functions as an n-gram model, where n represents the length of the word w. This approach dynamically captures contextual information at the character level using the variable n-gram model and extends this context into Viterbi decoding at the word level.

For example, suppose we have patterns "a", "ab", "ac", "bc" with counts $1, 2, 3, 4$, then we can calculate the probility of how possible "ab" could be a word as $\frac{2}{1+2+3}$ since it's longest recognizable prefix is "a", which is the common prefix of "a", "ab" and "ac". And for "bc", since there's no "b" in the patterns, it's calculated as $\frac{4}{1+2+3+4}$ because all of the patterns share common prefix ϵ denoting the empty string.

Ambiguities within words, such as "武汉市长" (the mayor of Wuhan city), might be wrongly segmented into "武汉市/长" (Wuhan city/long) by simple unigram models like Jieba, where both "武汉市" (Wuhan city) and "长" (long) are frequent patterns in Chinese. Our model addresses these situations more adeptly by leveraging contextual cues: "武汉" yields a lower perplexity in the context following "武", and similarly, "市长" (mayor) is more likely following "市" (city). Thus, in Viterbi decoding, our model can more accurately segment it as "武汉/市长" (the mayor of Wuhan city) rather than "武汉市/长" (Wuhan city is long).

We adopt a 2-tag system, where each decision either marks a boundary between words or allows the sequence to continue without a break. In Viterbi decoding, we traverse the sequence from beginning to end, updating the path with the maximum log probability for every prefix. Each position stores the optimal previous word boundary and its corresponding log probability. Once the path for the last prefix (the whole sequence) is updated, we can backtrack from it to the beginning of the sentence to determine all the word boundaries in the optimal path.

3.3 Time Complexity

The LLM-WS will be the most time-consuming part, primarily due to the inference efficiency of the LLM. Apart from that, every component of LLACA is robust and quick. PMI can be calculated in $\mathcal{O}(L^2 \cdot \log N)$, where L denotes the length of the word being analyzed, and N denotes the length of the raw text using a Suffix Array (Manber and Myers, 1993) built on the raw text. The construc-

Table 2: The comparison between Qwen1.5-7B-Chat and GPT-4.

Model	AS	CITYU	MSR	PKU	KWDLC	UD_JA	UD_KO	BEST	UD_TH
GPT-4	76.5	80.6	75.0	76.3	**79.7**	**78.7**	**44.9**	**76.3**	**75.1**
Qwen1.5-7B-Chat	**78.4***	**82.5***	**84.2**	**86.7**	52.7	49.8	36.8	8.9	21.3

tion of the AC automaton and the pattern matching are linear with respect to the sum of the lengths of the patterns and the raw text (Aho and Corasick, 1975), respectively. The time complexity of Viterbi decoding (Viterbi, 1967), which involves traversing the Directed Acyclic Graph (DAG) that represents all recognized patterns (as shown in Figure 5), is approximately $\mathcal{O}(N)$, depending on the number of patterns matched. More details are available in Appendix A.

4 Experiments

4.1 Experimental Setup

To evaluate the word segmentation capabilities of LLMs and the effectiveness of our proposed approach across different languages and scripts, we selected several open-sourced datasets for Chinese (AS, CITYU, MSR, PKU, Emerson, 2005), Japanese (KWDLC, Hangyo et al., 2012, UD_JA, Nivre et al., 2020), Korean (UD_KO, Nivre et al., 2020) and Thai (BEST, Kosawat et al., 2009, UD_TH, Nivre et al., 2020). We continued to use the token F-measure as our primary metric for evaluation, consistent with the methodology outlined in our preliminary experiments (3.1).

The sampling and testing procedures were conducted on the test sets. Except for the BEST dataset, which was randomly sampled from the training data, all other test datasets maintained their original splits. Our experiments primarily utilized the Qwen1.5 series of LLMs (Bai et al., 2023). We selected this series because it is reputed to excel in multilingual tasks, offering a wide range of parameters from 0.5B to 110B and easy to employ. The diversity of parameters enables us to explore the relationship between the models' parameter sizes and their comprehension abilities effectively. For further details about our experimental setup, please see Appendix A.

4.2 LLM-WS: Evaluating the Word Segmentation Capabilities of LLMs

Table 1 presents the results after one iteration and compares them with previous state-of-the-art unsupervised methods. Previous state-of-the-art unsupervised methods for Chinese word segmentation generally achieved scores around 80. However, Qwen1.5-7B-Chat, which was mainly pre-trained on Chinese corpus, has significantly advanced performance, reaching approximately 87. It is important to note that Qwen was primarily pre-trained on Simplified Chinese; consequently, it occasionally produces Simplified Chinese outputs when processing Traditional Chinese. To address this, texts are converted to Simplified Chinese using OpenCC[2], resulting in increased F-measure scores on the AS dataset from 67.9 to 78.4 and the CITYU dataset from 69.1 to 82.5.

As shown in Table 2, Qwen1.5-7B-Chat surpassed GPT models in tasks related to Chinese language processing, yet demonstrated less effectiveness in segmenting other languages. Qwen series occasionally produces translations into Simplified Chinese for words from other languages. Interestingly, the conversion from Traditional to Simplified Chinese can also be regarded as a form of "translation". This behavior does not imply a lack of comprehension by the Qwen1.5 models. Rather, it suggests that they can process how these "foreign" languages are structured and segmented. This kind of hallucination typically occurs in models where the primary pre-training corpora do not align with the target language they are tasked to segment. This observation underscores how the pre-training texts significantly influence LLMs' interpretation of prompts, akin to how a non-native English speaker might initially think in their mother tongue before translating thoughts into English. It highlights a shift from the traditional sequence of "segment first, comprehend later" to a more integrated approach where LLMs "comprehend first, segment later". This shift indicates a potential advancement in how language models process and understand text, integrating comprehension and segmentation in a more dynamic manner.

It's important to note that the Qwen1.5-32B-Chat model was released after the other models in the Qwen1.5 series as an intermediate option

[2] https://github.com/yichen0831/opencc-python

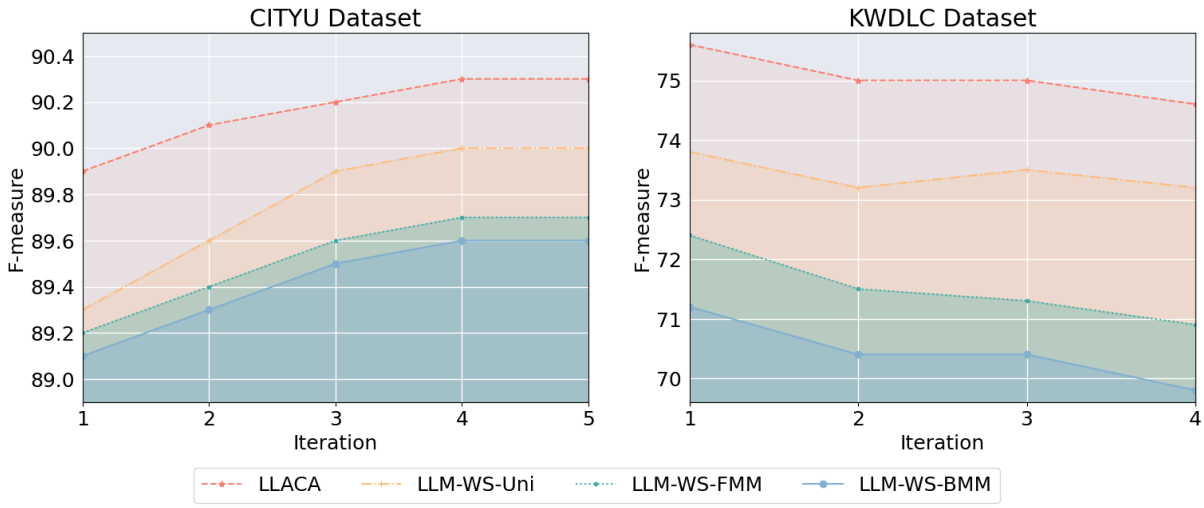

Figure 6: The evolution across iterations for four methods based on LLM-WS, starting from a baseline of 86.8 for the CITYU dataset and 72.3 for the KWDLC dataset with the Qwen1.5-110B-Chat model.

between the 14B and 72B models. Its architecture and training methodologies exhibit some variations from the other models in the series. These differences might explain why the Qwen1.5-32B-Chat model shows a slight decrease in performance on certain datasets compared to the 7B and 14B models. With the consistent architecture and an increase in parameters, F-measure generally improves across different languages, demonstrating that larger LLMs possess enhanced capabilities for generalization and comprehension in NLP. This trend indicates that the expansion in model capacity not only aids in handling more complex linguistic features but also improves adaptability to varied linguistic contexts.

4.3 From LLM-WS to LLACA

Although LLMs demonstrate capabilities in word segmentation, employing them for this specific NLP task is impractical due to the significant time and economic costs involved. From the results shown in Table 1 and 3, it is evident that out proposed approach LLACA not only offers a significantly faster inference speed but also maintains the statistical integrity of the patterns recognized by the LLM, often surpassing the LLM's performance. This improvement is largely attributed to LLACA's ability to eliminate hallucinations generated by the LLM.

Figure 6 illustrates the evolution of four lexicon-based methods inspired by Qwen1.5-110B-Chat over several iterations. **LLACA**: Our proposed method with variable n-gram possibility model, incorporating Viterbi decoding. **LLM-WS-Uni**: A straightforward unigram model, also incorporating Viterbi decoding. **LLM-WS-FMM**: The Forward Maximum Matching algorithm. **LLM-WS-BMM**: The Backward Maximum Matching algorithm.

Table 3: The comparison of the average inference time and F1 scores between Qwen1.5-14B-Chat and LLACA inspired by it over 4 iterations.

Model	**Time**↓ (MSR)	F ↑ (MSR)
Qwen1.5-14B-Chat	3.65h	85.9
LLACA	2.01s	87.7
Model	**Time**↓ (PKU)	F ↑ (PKU)
Qwen1.5-14B-Chat	2.85h	87.5
LLACA	1.80s	88.9

As the iterations progress, the four methods begin to plateau. The two statistical methods (LLACA, LLM-WS-Uni) consistently outperform the two greedy methods (LLM-WS-FMM, LLM-WS-BMM), as they account more thoroughly for global states rather than merely local matches. This outcome supports the premise that LLACA's enhancements to LLM outputs are not merely due to an expanded lexicon, but also because of the effective utilization of latent semantic information represented in statistical form. Among the statistical methods, our LLACA integreted with variable n-gram possibility model, as detailed in Equation 2, consistently excels over the simpler unigram model. This suggests its rationality in modeling the probabilities of natural language since ambiguity often occurs between similar patterns.

Besides, we observed that the performances

with Qwen1.5-110B-Chat was suboptimal on the Japanese dataset KWDLC. The presence of noise within the data patterns contributed to a slight degradation in performance as the model iterations progressed. Our LLACA was the least affected by the noisy data patterns and maintained a relatively good F-measure score overall. This suggests that LLACA is more robust to inputs with noises.

4.4 On OOV Handling

The tokenization of LLMs is based on UTF-8 code but not supervised vocabulary, and the LLM segments the words based on its comprehendion, there's no OOV issues of LLMs ideally. Our LLM-based approach LLACA actually offers advantages in handling OOV words: **1)** The dictionary can be dynamically supplemented for different domains through our unsupervised approach. **2)** This allows better domain adaptation and OOV handling compared to static dictionary approaches.

Considering that the most frequently used words are shared in multiple scenes, our method could handle the common part without forgetting them. Here we represent the comparison between SLM (Sun and Deng, 2018) and LLACA (inspired by Qwen1.5-7B) in Table 4.

Table 4: Comparison between SLM (Sun and Deng, 2018) and LLACA on OOV Handling for MSR, PKU, and CTB Datasets

Training-Model	Test		
	MSR	**PKU**	**CTB**
MSR-SLM-3	73.9	69.8	67.4
MSR-LLACA	**86.7**	**75.2**	**70.0**
PKU-SLM-3	70.8	76.6	69.2
PKU-LLACA	**78.2**	**87.7**	**72.9**
CTB-SLM-3	69.7	70.1	76.0
CTB-LLACA	**77.0**	**75.5**	**88.0**

Note that the SLM results listed here are from our re-train results based on its official implementation. To make a fair comparison, the unsupervised training process used only unsegmented test corpus the same as ours, thus the results might be different from what they are represented in the original paper of SLM. With ideally the same vocabulary, our approach shows consistent better performance across different domains.

To further substantiate the validity of the variable n-gram approach employed by our LLACA model, we have also compared with various modeling techniques. Detailed discussions on these comparisons can be found in the Appendix E.

5 Conclusion

In this paper, we explore the word segmentation capabilities of LLMs. We conclude that the task of word segmentation can serve as an effective measure of an LLM's ability to comprehend prompts and apply logical reasoning in natural languages. To fully harness the deep comprehension capabilities of LLMs, our proposed method LLACA integrates the rapid pattern recognition of the AC automaton with a novel variable n-gram model, surpassing previous benchmarks and setting new state-of-the-art unsupervised results.

Broader Impacts

Word segmentation is not a standalone task. The ambiguities need to be resolved through the context. Previous segmental models lacked the overall comprehension of the entire text, let alone multimodal comprehension. In the era of LLMs, LLM-WS takes a radically different approach, which can leverage long-range context and even multimodal associations. This emergent ability has transformed word segmentation from a pure statistical machine learning problem to a new paradigm. Building on the capabilities of LLMs, we propose LLACA, which can infer faster and perform better than previous unsupervised word segmentation methods. We hope our work will inspire more LLM-based NLP researches and applications.

Limitations

Now that word segmentation has evolved into a comprehension task, the upper limit may lie in the inherent inconsistencies of word segmentation itself. Some segmentation results from LLMs may accurately reflect the language understanding, yet they may not align with the traditional "golden standard" annotations. To better assess the language comprehension capabilities of LLMs in NLP tasks, new evaluation standards should be developed to align with the current paradigm shift. The existing standards, which were designed for previous segmentation models, may no longer adequately capture the nuanced understanding exhibited by LLM-based approaches.

References

Alfred V Aho and Margaret J Corasick. 1975. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340.

Galen Andrew. 2006. A hybrid Markov/semi-Markov conditional random field for sequence segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 465–472, Sydney, Australia. Association for Computational Linguistics.

J-I Aoe. 1989. An efficient implementation of static string pattern matching machines. *IEEE Transactions on Software Engineering*, 15(8):1010–1016.

Miguel Arevalillo-Herráez, Pablo Arnau-González, and Naeem Ramzan. 2022. On adapting the DIET architecture and the rasa conversational toolkit for the sentiment analysis task. *IEEE Access*, 10:107477–107487.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany. Association for Computational Linguistics.

Bob Carpenter. 2006. Character language models for chinese word segmentation and named entity recognition. In *Proceedings of the Fifth Workshop on Chinese Language Processing, SIGHAN@COLING/ACL 2006, Sydney, Australia, July 22-23, 2006*, pages 169–172. Association for Computational Linguistics.

Paulo Roberto Massa Cereda, Newton Kiyotaka Miura, and João José Neto. 2018. Syntactic analysis of natural language sentences based on rewriting systems and adaptivity. In *The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT 2018) / Affiliated Workshops, May 8-11, 2018, Porto, Portugal*, volume 130 of *Procedia Computer Science*, pages 1102–1107. Elsevier.

Miaohong Chen, Baobao Chang, and Wenzhe Pei. 2014. A joint model for unsupervised chinese word segmentation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 854–863. ACL.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015. Gated recursive neural network for Chinese word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1744–1753, Beijing, China. Association for Computational Linguistics.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. 2009. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2012. Building a diverse document leads corpus annotated with semantic relations. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 535–544, Bali, Indonesia. Faculty of Computer Science, Universitas Indonesia.

Liqi He, Zuchao Li, Xiantao Cai, and Ping Wang. 2024. Multi-modal latent space learning for chain-of-thought reasoning in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18180–18187.

Changning Huang and Hai Zhao. 2007. Chinese word segmentation: A decade review. *Journal of Chinese Information Processing*, 21(3):8–20.

Hongjiang Jing, Zuchao Li, Hai Zhao, and Shu Jiang. 2021. Seeking common but distinguishing difference, a joint aspect-based sentiment analysis model. *arXiv preprint arXiv:2111.09634*.

Dan Jurafsky and James H Martin. 2014. Speech and language processing. vol. 3.

Donald E Knuth, James H Morris, Jr, and Vaughan R Pratt. 1977. Fast pattern matching in strings. *SIAM journal on computing*, 6(2):323–350.

Krit Kosawat, Monthika Boriboon, Patcharika Chootrakool, Ananlada Chotimongkol, Supon Klaithin, Sarawoot Kongyoung, Kanyanut Kriengket, Sitthaa Phaholphinyo, Sumonmas Purodakananda, Tipraporn Thanakulwarapas, et al. 2009. Best 2009: Thai word segmentation software contest. In *2009 Eighth International Symposium on Natural Language Processing*, pages 83–88. IEEE.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si. 2018. A unified syntax-aware framework for semantic role labeling. In *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, pages 2401–2411.

Zuchao Li, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2020. Explicit sentence compression for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8311–8318.

Pierre Magistry and Benoît Sagot. 2012. Unsupervized word segmentation: the case for Mandarin Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–387, Jeju Island, Korea. Association for Computational Linguistics.

Udi Manber and Gene Myers. 1993. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 22(5):935–948.

Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 591–598. Morgan Kaufmann.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 100–108, Suntec, Singapore. Association for Computational Linguistics.

Yasmin Moslem. 2024. Language modelling approaches to adaptive machine translation. *CoRR*, abs/2401.14559.

Mahdi Namazifar, Alexandros Papangelis, Gökhan Tür, and Dilek Hakkani-Tür. 2021. Language model is all you need: Natural language understanding as question answering. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 7803–7807. IEEE.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. *arXiv preprint arXiv:2004.10643*.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 562–568, Geneva, Switzerland. COLING.

Jay M. Ponte and W. Bruce Croft. 2017. A language modeling approach to information retrieval. *SIGIR Forum*, 51(2):202–208.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 253–262. The Association for Computer Linguistics.

Xu Sun, Yao-zhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. A discriminative latent variable chinese segmenter with hybrid word/character information. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, pages 56–64. The Association for Computational Linguistics.

Zhiqing Sun and Zhi-Hong Deng. 2018. Unsupervised neural word segmentation for chinese via segmental language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4915–4920. Association for Computational Linguistics.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2004. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 1385–1392.

Kei Uchiumi, Hiroshi Tsukahara, and Daichi Mochihashi. 2015. Inducing word and part-of-speech with pitman-yor hidden semi-markov models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1774–1782. The Association for Computer Linguistics.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Chong Wang, Yining Wang, Po-Sen Huang, Abdelrahman Mohamed, Dengyong Zhou, and Li Deng. 2017. Sequence modeling via segmentations. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3674–3683. PMLR.

Chunqi Wang and Bo Xu. 2017. Convolutional neural network with word embeddings for Chinese word segmentation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 163–172, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Hanshi Wang, Jian Zhu, Shiping Tang, and Xiaozhong Fan. 2011. A new unsupervised approach to word segmentation. *Comput. Linguistics*, 37(3):421–454.

Lihao Wang and Xiaoqing Zheng. 2022. Unsupervised word segmentation with bi-directional neural language model. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(1):1–16.

Xingyu Yan, Xiaofan Xiong, Xiufeng Cheng, Yujing Huang, Haitao Zhu, and Fang Hu. 2021. Hmm-bimm: Hidden markov model-based word segmentation via improved bi-directional maximal matching algorithm. *Comput. Electr. Eng.*, 94:107354.

Yao Yao, Zuchao Li, and Hai Zhao. 2023. Beyond chain-of-thought, effective graph-of-thought reasoning in language models. *arXiv preprint arXiv:2305.16582*.

Yushi Yao and Zheng Huang. 2016. Bi-directional LSTM recurrent neural network for chinese word segmentation. In *Neural Information Processing - 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16-21, 2016, Proceedings, Part IV*, volume 9950 of *Lecture Notes in Computer Science*, pages 345–353.

Zhuosheng Zhang, Hai Zhao, Kangwei Ling, Jiangtong Li, Zuchao Li, Shexia He, and Guohong Fu. 2019. Effective subword segmentation for text comprehension. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11):1664–1674.

Hai Zhao, Deng Cai, Changning Huang, and Chunyu Kit. 2019. Chinese word segmentation: Another decade review (2007-2017). *CoRR*, abs/1901.06079.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An improved Chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165, Sydney, Australia. Association for Computational Linguistics.

Hai Zhao and Chunyu Kit. 2008. An empirical comparison of goodness measures for unsupervised chinese word segmentation with a unified framework. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

A Experimental Details

A.1 Algorithm

It is feasible to use the LLACA automaton independently of the LLM after its initial construction. This flexibility allows for manual additions to the vocabulary or for continued refinement and "distillation" from existing LLM outputs. As LLMs are exceptionally effective at exploring and identifying new vocabulary—acting as advanced "word explorers"—the issue of handling out-of-distribution data is significantly mitigated. We can always tailor the LLACA to be domain-specific, thus maintaining its effectiveness across varying data distributions. Algorithm 1 describes the overall procedure including vocabulary construction and word segmentation.

Furthermore, the design of LLACA is inherently extensible, with each recognized pattern functioning as a weighted edge within the automaton. This structure facilitates the integration of additional patterns and rules, such as regular expressions for identifying URLs or other specialized formats. By interpolating the weights assigned to these new patterns, LLACA can be adapted to recognize and process a broader array of textual features, enhancing its utility and applicability.

A.2 Datasets

Chinese We utilized four standard datasets from the SIGHAN Bakeoff 2005 (Emerson, 2005), widely recognized in computational linguistics research: AS, CITYU, MSR, and PKU. The AS and CITYU datasets contain Traditional Chinese texts, while MSR and PKU are composed of Simplified Chinese texts. These datasets provide a robust foundation for evaluating segmentation performance across various forms of Chinese script.

Japanese For Japanese, we employed the Kyoto University Word Dependency Corpus (KWDLC) (Hangyo et al., 2012) and the Universal Dependencies (UD) Japanese GSD treebank (Nivre et al., 2020), which are instrumental in studying segmentation in scripts without clear delimiters.

Korean For Korean, we employed the Universal Dependencies (UD) Korean GSD treebank (Nivre et al., 2020). Additionally, we deleted the spaces in the original test dataset.

Thai For Thai, we utilized the BEST dataset (Kosawat et al., 2009) and the UD Thai PUD treebank (Nivre et al., 2020). Additionally, we created a subset by randomly selecting 1,000 sentences from the BEST-Novel dataset's training data. All delimiters were removed from this subset to prepare it for use as both training and testing data.

Statistics and Licenses for Datasets Table 5 provides a summary of the datasets used for evaluation, while Table 6 details the licenses associated with each dataset.

Except for BEST was a subuset contained 1,000 sentences randomly sampled with random seed 17 by `numpy.random.choice` from the BEST-Novel training dataset, all other test datasets maintained

Table 5: The statistics overview of the datasets used in the evaluation across different languages: Chinese (ZH), Japanese (JA), Korean(KO), and Thai (TH).

Language	Dataset	Test Size (KB)	Total Tokens	Total Chars	Unique Tokens	Unique Chars
ZH	AS	603	122610	197681	18811	3707
	CITYU	196	40936	67690	9001	2702
	MSR	547	106873	184355	12923	2838
	PKU	497	104372	172733	13148	2934
JA	KWDLC	194	35869	64905	6144	1821
	UD_JA	62	13034	21322	3568	1494
KO	UD_KO	100	11677	32742	7102	1125
TH	BEST	328	31697	112261	3804	102
	UD_TH	282	22322	96161	4047	134

their original splits. We only used their test datasets for LLM-WS and testing.

Table 6: Licenses of the datasets.

Dataset	License
AS	Research Purpose
CITYU	Research Purpose
MSR	Research Purpose
PKU	Research Purpose
KWDLC	Research Purpose
UD_JA	CC BY-SA 4.0
UD_KO	CC BY-SA 4.0
BEST	CC BY-NC-SA 3.0
UD_TH	CC BY-SA 3.0

A.3 Experimental Environment

For close-sourced models GPT-3.5-Turbo and GPT-4, we employed them by API. And for other open-sourced models, we initialized them from pre-trained checkpoints and employed them on 8 A100-SXM80GB GPUs.

A.4 Parameters

We set the top ratio to a conservative level of 0.99, allowing LLACA to incorporate a broader range of words into the analysis.

B Baselines

The detail of baseline models is following:

- nVBE (Magistry and Sagot, 2012): a system utilizing Normalized Variation of Branching Entropy.
- NPY-n (Mochihashi et al., 2009): a nested n-gram hierarchical Pitman-Yor language model, where Pitman-Yor spelling model is embedded in the word model.
- Joint (Chen et al., 2014): the "HDP+HMM" model initialized with nVBE model.
- SLM-n (Sun and Deng, 2018): the first neural model for unsupervised CWS, where n denotes the maximum word length.
- SGB-A, SGB-C (Wang and Zheng, 2022): a model maximizing the generation probability of a sentence given its all possible segmentation, where A and C denote 2 different decoding algorithms.
- PYHSMM (Uchiumi et al., 2015): a nonparametric Bayesian model for joint unsupervised word segmentation and part-of-speech tagging from raw strings.

C Analysis of the Limits of Unsupervised Word Segmentation

As indicated by (Huang and Zhao, 2007; Zhao and Kit, 2008), standard inconsistencies occur across different datasets for word segmentation. Therefore, it is essential to consider consistency across these datasets as the upper limit for evaluating unsupervised segmentation methods. We selected three Simplified Chinese datasets and pre-trained segmental models on each to replay the experiment in (Huang and Zhao, 2007). The model architecture is adapted from (Wang and Xu, 2017) and we utilizes pre-trained models provided by HanLP[3].

Let the F-measure of each model on its respective test dataset be denoted as F_0. Typically, supervised models perform optimally on their corresponding test sets but exhibit diminished performance on others. Hence, we use F_0 as a normalization factor to measure consistency across datasets.

[3] https://github.com/hankcs/HanLP/tree/master

Algorithm 1 Word Segmentation with LLACA

Input: Raw text data
Output: Segmented text
procedure LLACA(text)
    Randomly shuffle the raw text into batches
    for each batch **do**
        Get patterns from LLM-WS
        Apply PMI filtering to refine patterns
        Add filtered patterns to LLACA
    end for
    Pre-process with normal patterns like number, alpha and symbols
    Initialize Viterbi probabilities based on patterns in LLACA using Equation 2
    Prepare a table to track paths and their probabilities
    for each character index i in text **do**
        for each state s representing a possible pattern ending at i **do**
            Calculate the highest probability path ending in s using:
            $\max_{s'} \log P(s' \to s) + \log P(s')$
            where s' is a state leading to s and P are the transition probabilities
            Store this path if it has the highest log probability for s
        end for
    end for
    Backtrack from last character to find the path with the highest probability
    return Segmented text based on the best path
end procedure

The values of F_0 are 95.4 (CTB), 97.1 (MSR), and 95.5 (PKU). Each F-measure was normalized by the F_0 of its training dataset, achieving a consistency score of 1.0 for each model on its test dataset.

Table 7: Consistency rate among Simplified Chinese datasets CTB, MSR and PKU

Test	Trainig		
	CTB	**MSR**	**PKU**
CTB	1.000	0.865	0.944
MSR	0.871	1.000	0.947
PKU	0.933	0.880	1.000

As shown in Table 7, the lowest consistency rate observed is 86.5%, while the highest is 94.7%, with an average consistency of 93.8%. Some previous studies have regarded that the lowest consistency rate is the ceiling for unsupervised word segmentation methods (Sun and Deng, 2018; Uchiumi et al., 2015). We propose that it is more reasonable to estimate the ceiling as the average consistency. Utilizing Qwen1.5-110B-Chat, our LLACA achieved a 90.3 F-measure on the CITYU dataset.

Aside from the inconsistency of segmentation criteria, most traditional unsupervised word segmentation methods are based on boosting. However, our approach is based on the semantic understanding capabilities of LLMs. Therefore, our method has re-explored the upper limit of unsupervised word segmentation, which also demonstrates that our approach has ushered in a new era for the development of unsupervised word segmentation.

D Aho-Corasick Automaton

Once the Trie is assembled, failure links are established. Each failure link at a node connects to the longest proper suffix of the string at that node, which also serves as a prefix for another pattern in the Trie. If no such suffix exists, the link reverts to the root. This is analogous to the "failure function" in the Knuth-Morris-Pratt (KMP) string-matching algorithm (Knuth et al., 1977), but Aho-Corasick extends this idea to work efficiently for multiple patterns.

Figure 3 illustrates the AC automaton's structure, showcasing failure links in red and final states with double circles, though some transitions may be omitted for clarity. The process begins with an input sequence that progresses through the Trie. If a mismatch occurs, such as when at the state "she" and the next input is "r" without a corresponding edge, the automaton utilizes failure links to backtrack until a valid node with the "r" edge is found or until it returns to the root. When the automaton reaches the state "her", not only is the pattern "her" itself recognized but the state of its failure pointer is also included. This forms part of a recursive process: matching a state involves sequentially matching the state of its failure pointer until it traces back to the root node, which represents the absence of further matches, denoted as ϵ.

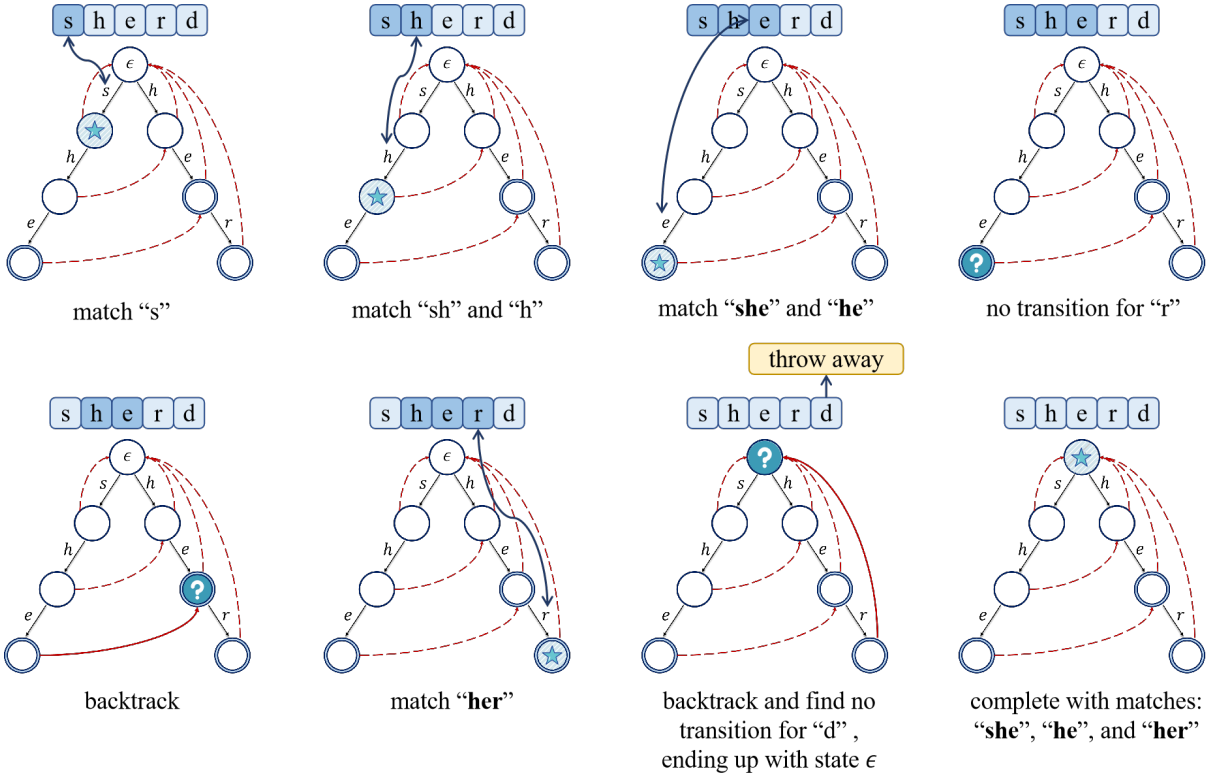

Figure 7: The process of how "sherd" matches patterns "she", "he", and "her" by transitions on an AC automaton.

For further clarification, Figure 7 illustrates the matching process for the string "sherd", identifying the substrings "she", "he", and "her". The elements highlighted in dark blue represent both the longest pattern prefix that the current state can match, and the minimal suffix information necessary for subsequent matches. This setup can be visualized as a "sliding window" that moves from left to right across each character. During normal transitions, this sliding window accordingly steps to the right. Conversely, during backtracking, the state transitions via the failure pointer, effectively discarding any irrelevant left-side components. Note that in actual implementations, the failure links in AC automata are primarily used during the construction phase and the match phase. Once the automaton is constructed, these failure links are often replaced by virtual transitions that directly lead to the correct states like Algorithm 2. This optimization streamlines the matching process, enhancing efficiency by reducing unnecessary transitions.

The double-array Trie (Aoe, 1989), markedly enhances the efficiency of space utilization in Trie data structures. This innovation reduces the memory footprint traditionally associated with Trie implementations. As a result, the double-array-based Aho-Corasick automaton emerges as an almost ideal solution for applications involving multiple pattern matching due to its optimal balance of space efficiency and performance.

E Further Validations for LLACA

E.1 Probability Modeling

LLACA employs a variable n-gram approach to model probabilities at a global level. This method not only captures local dependencies but also considers global contextual information. Let p_i represent the probability that the model assigns to the segmentation of the i-th sentence s_i. The perplexity of one sentence can be defined as follows, where n_i denotes the length of s_i:

$$\mathrm{PPL}(s_i) = \sqrt[n_i]{\frac{1}{p_i}} \tag{3a}$$

$$= \exp\left(-\frac{\log p_i}{n_i}\right) \tag{3b}$$

For the total texts, we still adopt geometric mean. The perplexity for $\mathbf{t} = [s_1, s_2, \cdots, s_k]$ could be

Table 8: Starting from an "ordinary dictionary" trained on the *People's Daily* corpus provided by Jieba, further expanded by additional dictionaries extracted from raw texts in the MSR dataset using Qwen1.5-7B to Qwen1.5-72B.

	Ord.	+Qwen1.5-7B	+Qwen1.5-14B	+Qwen1.5-32B	+Qwen1.5-72B
Ours	**85.5**	**86.5**	**86.8**	**86.8**	**87.0**
Jieba	82.7	80.3	80.7	80.4	80.8

measured as below:

$$\mathrm{PPL}(\mathbf{t}) = \sqrt[k]{\mathrm{PPL}(s_1)\,\mathrm{PPL}(s_2)\cdots\mathrm{PPL}(s_k)} \tag{4a}$$

$$= \sqrt[k]{\Pi_{i=1}^{k} \exp\left(-\frac{\log p_i}{n_i}\right)} \tag{4b}$$

$$= \exp\left(\frac{\sum_{i=1}^{n} -\frac{\log p_i}{n_i}}{k}\right) \tag{4c}$$

E.2 Baselines

Tools mentioned in the discussion are available in GitHub. Details of them are below:

- Jieba[4]: Jieba is a highly popular Chinese text segmentation tool known for its ease of use and versatility. Jieba also allows for custom dictionary integration, making it adaptable for specific vocabularies or industry terms.
- MeCab[5]: MeCab is a sophisticated morphological analyzer for Japanese text, implemented in C++ and employing Conditional Random Fields (CRF) as its core algorithm.
- Komoran (in KoNLPy[6]): As part of the KoNLPy suite, Komoran is tailored for Korean text segmentation. It is particularly noted for its accuracy in analyzing formal and well-structured documents. Komoran is suitable for academic and professional applications where precision is crucial.
- AttaCut (in PyThaiNLP[7]): A modern tool designed specifically for Thai, AttaCut is embedded in PyThaiNLP project. It uses deep learning models, particularly CNNs, to segment Thai text, which is known for its absence of clear word boundaries.
- NewMM (in PyThaiNLP): Another engine embedded in PyThaiNLP, NewMM is a multi-dictionary-based maximizer matching algorithm that efficiently handles WS in Thai. It's designed to be fast and is the default engine in PyThaiNLP due to its robustness in general-purpose applications.

E.3 Starting from an Ordinary Dictionary

Starting with an "ordinary dictionary" trained on the *People's Daily* corpus provided by Jieba, further expanded by additional dictionaries extracted from raw texts in the MSR dataset using Qwen1.5-7B to Qwen1.5-72B. The results, evaluated using F1 scores on the MSR dataset, are presented in Table 8.

E.4 Comparisons with "Off-the-Shelf" Tools

Table 9 presents comparisons with other off-the-shelf tools in Japanese, Korean and Thai. While these results are not directly comparable, they serve as an indicator of the limitations inherent in supervised methods and the advantages of our unsupervised approach. Supervised methods often struggle with out-of-domain (OOD) and out-of-vocabulary (OOV) issues. However, in practical scenarios, labeled domain-specific data is not always abundantly available. Utilizing LLMs as our "word explorer", similar to human cognitive processes, ideally eliminates concerns related to OOD and OOV. Many supervised methods have not transitioned into practical tools due to concerns over efficiency and the ability for customization. Here, we demonstrate the transferability and high efficiency of LLACA, underscoring its capability to be effectively implemented in real-world applications. We are confident in and committed to the potential of LLACA.

Table 10 shows the perplexity comparison between LLM-WS-Uni and LLACA. On four datasets, LLACA achieved higher F-measure and lower perplexity compared to LLM-WS-Uni. Higher F-measure and lower perplexity represent a higher probability of the word segmentation results, implying that the word segmentation results

[4] https://github.com/fxsjy/jieba
[5] https://github.com/taku910/mecab
[6] https://github.com/konlpy/konlpy
[7] https://github.com/PyThaiNLP/pythainlp

Table 9: Compared with "off-the-shelf" tools for Japanese, Korean, and Thai languages, † indicates that the training set may overlap with the test set used here. * denotes that we added the same training vocabulary as ours to ensure a more equitable comparison.

	MeCab		Komoran		AttaCut		NewMM		Ours	
	F1	Time (s)	F1	Time (s)	F1	Time (s)	F1	Time (s)	F1	Time (s)
KWDLC	88.8	0.06							91.9	0.67
UD_KO			27.9*	0.58					51.2	0.38
BEST					97.0†	14.09	75.3*	0.31	93.5	1.09

Table 10: Perplexity (the lower the better) and F-measure (the larger the better) on datasets of different languages. LLM-WS-Uni and LLACA's construction were both conducted on GPT-4.

	CITYU		MSR	
Model	F↑	ppl.↓	F↑	ppl.↓
LLM-WS-Uni	83.7	121	84.1	94
LLACA	**84.1**	**42**	**84.2**	**28**
	KWDLC		BEST	
Model	F↑	ppl.↓	F↑	ppl.↓
LLM-WS-Uni	82.7	51	72.1	14
LLACA	**83.4**	**21**	**72.8**	**8**

are more reasonable.

F Discussions

F.1 With LLMs, Why We Still Need Word Segmentation (WS)

BPE tokenizers, especially the one used in OpenAI's Tiktoken (also employed by Qwen-1.5), operate at a *byte-level* and do not inherently understand semantic boundaries. This means that while models like GPT and Qwen-1.5 perform impressively on many tasks, their understanding is based on statistical co-occurrence rather than semantic comprehension. The impressive performance of these models can largely be attributed to their well-designed architectures and extensive pre-training.

However, this comes at a cost. The high time and space complexity of inference with such large models can be prohibitive, particularly in environments where computational resources are limited. Furthermore, according to the "No Free Lunch Theorem", these models may still lag behind in domain-specific tasks compared to specialized NLP tools. Additionally, smaller models are often prone to hallucinations due to their reduced capacity and generalist training.

Given these considerations, the task of WS remains critically important for several reasons:

- **Efficiency**: Many NLP tasks require high computational efficiency. Fast and effective WS tools can provide essential semantic information more quickly and with fewer resources than LLMs. For example, some rule-based methods could greatly benefit from this.
- **Accessibility**: Not all researchers and developers have the means to deploy and maintain LLMs like Qwen efficiently. By continuing to develop and improve WS techniques, we ensure that robust, less resource-intensive solutions are available, keeping the field of NLP inclusive and versatile.
- **Reliability**: Effective segmentation reduces errors in further processing steps, such as parsing and translation, ensuring that the output is both accurate and contextually appropriate. This is crucial in professional settings where precision is paramount, such as legal and medical document analysis.

In summary, while large pre-trained models offer broad capabilities, WS tasks play a crucial role in achieving high accuracy and efficiency in specific applications, ensuring that CJK NLP technology remains accessible and practical across a diverse range of use cases.

F.2 Why We Still Need Unsupervised Word Segmentation

Generally, we consider unsupervised methods from these aspects:

- **Data Availability and Cost**: Acquiring labeled data is frequently costly and time-consuming. In many practical scenarios, such data may not even be available. Unsupervised learning, on the other hand, does not require labeled inputs and can be applied directly to raw data. This makes it particularly valuable in situations where data labeling is impractical or too expensive.

Algorithm 2 Calculate fail links and virtual transitions for nodes in an AC automaton

```
function GET_TRANSITIONS
    Initialize an empty queue Q
    for v ← children of the root (ϵ) do
        fail(v) ← ϵ                                   ▷ Set initial fail state to ϵ
        Enqueue v into Q
    end for
    while not Q.isEmpty() do
        u ← Q.dequeue()
        for i ← possible transitions from u do
            v ← child(u, i)
            if v ≠ ϵ then
                fail(v) ← child(fail(u), i)           ▷ Update the failure pointer
                Enqueue v into Q
            else
                child(u, i) ← child(fail(u), i)       ▷ Set virtual transition
            end if
        end for
    end while
end function
```

- **Pattern Discovery**: Unsupervised learning excels at discovering hidden patterns and structures in data that are not initially evident. For instance, clustering algorithms can reveal intrinsic groupings and structures within the data that supervised methods might overlook because they focus solely on the target outcomes defined by the labeled data.
- **Flexibility and Adaptability**: In dynamic environments where data distributions change over time, supervised models may require retraining with new labeled data, which can be both costly and impractical. Unsupervised learning models, can adapt to changes in input data without needing completely new labels.

Regarding the WS task, we introduce LLACA as a practical solution to maintain the WS capabilities of LLMs for actual use. This approach can significantly benefit various NLP tasks and supervised models by providing semantic information derived from words. Furthermore, LLACA naturally adapt to changes in domain and era, making it a versatile tool in the evolving landscape of NLP.