Tokenization via Language Modeling: the Role of Preceding Text

Rastislav Hronsky, Emmanuel Keuleers
Jheronimus Academy of Data Science, Tilburg University
Sint Janssingel 92 5211 DA ’s-Hertogenbosch, Warandelaan 2 5037 AB Tilburg
r.hronsky@tue.nl, E.A.Keuleers@tilburguniversity.edu

Abstract

While language models benefit immensely from their capacity to model large context (i.e., sequence of preceding tokens), the role of context is unclear in text tokenization, which is, in many cases, language model-driven to begin with. In this paper, we attempt to explore the role in three different writing systems and using three different text tokenization strategies (word-based, Morfessor, and BPE). In the first experiment, we examined how the size of context used for predicting the next token affects the ranking of the segmentation strategies i.t.o. language model surprisal. This effect was very writing system specific: minimal in case of English, and rank-reversing due to increased context size and token granularity in case of Turkish and Chinese. In the second experiment, we examined how context alters segmentation hypotheses when using language models to identify word boundaries. In this case, the effect was subtle: using context-aware, rather than context-free segment scores improved boundary recognition accuracy by up to 0.5%, once baseline effects were exploited.

Keywords: language modeling, tokenization, word segmentation

1. Introduction

The most basic unit of computer-stored written language is typically the character. Despite that neural network based systems are capable of taking characters as input, it is still common practice to divide the signal into linguistically more meaningful chunks (i.e., tokens). Most writing systems include conventions, such as whitespace and punctuation, that can help with segmentation. However, relying on these conventions to tokenize text is fragile: (1) there are many writing systems with different conventions, (2) even if explicit cues for word separation are available, further division, for instance of compound words, remains problematic, and (3) the noisiness and openness of natural language make dictionary-based string matching unreliable.

Therefore, modern systems pre-process text via pipelines that, in addition to using manually described rules, employ statistical segmentation, which is robust, language independent, and data driven. The idea is similar to how speech segmentation is described in studies on spoken language acquisition: a standalone token (word) is one where, within the boundaries, the regularity (mutual information) between neighboring elements (phonemes) is disproportionately stronger than at the boundaries (Saffran et al., 1996a).

This principle can be formulated more generally as a search for the set of segments that, by scoring the probability of each segment in isolation, maximizes the sequence generation probability (discounting any between-segment dependencies). In this form, it has been adopted as the decoding strategy for many text tokenization implementations (Creutz and Lagus, 2005; Virpioja et al., 2013; Sennrich et al., 2016; Kudo, 2018; Kudo and Richardson, 2018).

While these systems produce satisfactory tokens for their intended purposes, there is a lack of attention to the role of context in tokenization in natural language processing research. This is surprising, because statistical text segmentation is an application of probabilistic language models and modern language models have a capacity to model context extending to thousands of preceding tokens.

As a simple example of how context can affect segmentation, consider the sequence ’ishe’. Given a unigram model, the segmentation ’i-she’ may be optimal because ’i’ is a high frequency token in English. However, since ’is-he’ co-occurs often, the increased probability of ’he’ in the context of ’is’ may result in an overall higher probability of the ’is-he’ segmentation according to a bigram model. This way, it is conceivable how such a context-free (unigram) and context-sensitive (bigram) approach to segmentation would be in disagreement with each other.

Research on language acquisition confirms that context can affect segmentation. Language learners who make the independence assumption, hence ignore context, tend to identify words less accurately than ones that include the dependency to the preceding word (Goldwater et al., 2009).

A further indication of context utility comes from word segmentation research on writing systems without explicit word boundary notation: several systems employ contextual features to improve word segmentation performance (Meknawin, 1995; Kudo, 2006; Huor et al., 2004a; Durrani and Hussain, 2010).

To the best of our knowledge, the extent to which unigram and higher-order n-gram segmentation models correctly recover linguistic units, e.g., words, has not been studied in detail. Our first research question is:

(1) *How does using a higher-order language model (i.e., bigram or trigram as opposed to unigram) affect the performance of statistical word segmentation?*

To answer this question, we simulate word segmentation by deleting explicit word boundary notation and testing how well a uni-, bi-, and trigram model re-discover the reference word boundaries. We discuss the effects of increased n-gram model order in the context of merely using a more representative language model (i.e., inferred using a larger body of text), a baseline effect.

Given a particular segmentation of a corpus, we can derive a language model based on it and compute the average language model surprisal, a metric reflecting segmentation optimality. This is a common way to intrinsically assess segmentation, both between competing segmentation algorithms and within the decoding process of a single segmentation method, and it is the basis for our second, more general research question:

(2) *How does changing the order of the language model change the assessment of surprisal-based segmentation optimality?*

To answer this question, we present simulations examining the extent to which a particular corpus segmentation, e.g., a reference word segmentation, Morfessor, or BPE segmentation, ranks as consistently optimal (i.t.o. bits-per-sentence) across language models set to include increasingly long dependencies. If the ranking stays constant, this indicates a weak role of context.

We conducted both experiments with text in English (a morphologically poor language), Turkish (a very agglutinative language), and Chinese (a language with a logographic writing system). The corpus used was based on movie subtitles and aligned such that for each language it contains subtitles for the same set of movies.

One difficulty that we faced during this research was the lack of accessible scientific libraries to perform higher order segmentation. Therefore, we describe the process in Section 3, and also publicly release the code as a Python package [1] that we used to solve first, second, and third order sequence segmentation problems. The package is built on top of NetworkX (Hagberg et al., 2008), a popular network analysis library, providing direct access to a wide range of tools that can be used to manipulate and visualize the segmentation problems.

2. Related Work

2.1. Word Recognition in Spoken Language

While certain acoustic features help with word segmentation in speech recognition (Jusczyk et al., 1993; Mattys et al., 1999), the exact mechanics are non-trivial: the signal is often noisy and contains hardly any explicit word boundary signature (Cole et al., 1980; Reddy, 1976). Tasking humans with word identification from spectrograms of continuous speech is problematic itself (Klatt and Stevens, 1973). Unsurprisingly, modern automatic speech recognition (ASR) omit manual feature engineering and learn to transcribe speech to words end-to-end (Anusuya and Katti, 2009; Hannun et al., 2014; Amodei et al., 2016).

The challenges associated with ASR naturally transfer to language acquisition research: how do language learners identify words? A prominent finding from this literature is that the expectation of a phoneme pair at a word boundary to have a lower transition probability than one within a word is a reliable cue for word segmentation (Saffran et al., 1996a,b). As a result, the idea of exploiting statistical properties to segment speech into words has gained prominence (Brent, 1999; Venkataraman, 2001; Batchelder, 2002). Relatedly, computational models capitalizing on regularities *between* words, in addition to within words, improve word boundary recognition (Goldwater et al., 2009), especially by reducing undersegmentation (falsely omitting word boundaries).

2.2. Writing Systems without Whitespace

Word segmentation is an important topic for languages employing writing systems without explicit word delimiters (e.g., Chinese, Japanese, Thai or Khmer). Using word units was mainly needed for efficient functioning of traditional information retrieval systems (Nie et al., 1996; Chen et al., 1997; Leong and Zhou, 1997; Foo and Li, 2004). Simplifying matters somewhat, the word segmentation methods related to statistical segmentation were based on: (1) variants of *dictionary* based string matching for Chinese (Chen and Liu, 1992; Sproat and Emerson, 2003), Thai (Rarunrom, 1991; Virach, 1993), Khmer (Bi and Taing, 2014a), Japanese (Sato, 1999); (2) *statistical* approaches for Chinese (Sproat and Shih, 1990; Ge et al., 1999; Sun et al., 1998), Thai (Pornprasertkul, 1994; Meknawin, 1995), Khmer (Huor et al., 2004a), Japanese (Matsumoto et al., 2000; Kudo, 2006); (3) pipelines involving the statistics and several other rules and features (Meknavin et al., 1997). However, recent research questions the necessity for word segmentation by arguing that modern models based on

[1] https://github.com/hrasto/segmentgraph

characters, instead of words, generalize better and reduce overfitting (Li et al., 2019).

2.3. Vocabularies in modern NLP

In English-centric research, the traditional unit – word or lemma – was just about rejected in favor of subwords once neural networks became mainstream (Mikolov et al., 2012). This shift was mainly motivated by conveniences such as reduction of vocabulary size and robustness in handling out-of-vocabulary situations. Discounting linguistic rigor and aiming for robust engineering, several algorithms were developed to segment text into short subword units. The methods were typically based on a greedy compression algorithm: byte-pair encoding (BPE) (Gage, 1994; Sennrich et al., 2016), and its derivatives (Schuster and Nakajima, 2012; Kudo and Richardson, 2018).

Several studies further examined the effects of segmentation on language modeling and related tasks. Huck et al. (2017) found that using linguistically informed segmentation (e.g., compound splitting, prefix splitting, etc.) can improve machine translation (MT) performance over purely compression-based segmentation. Domingo et al. (2019) concluded that, while segmentation affects MT performance, there is no clear winner in terms of algorithms, as performance varies across language pairs. In language modeling experiments, Liu et al. (2019) found that there was a small advantage in using BPE-derived tokens from characters rather than bytes, and Gallé (2019) report that tokenizers producing fewer (thus longer) segments perform better.

Lastly, research suggests that there are advantages in using morphologically aligned subwords. Bostrom and Durrett (2020) compared segmentation produced by BPE to the Unigram method (Kudo, 2018), and found the latter to produce more morpheme-like tokens and ultimately outperforming BPE. Similarly, Park et al. (2021) report advantages in using segmentations produced by Morfessor (Creutz and Lagus, 2005), an unsupervised morphological segmentation system, over the BPE-based segmentations. Both methods, Unigram (Kudo, 2018) and Morfessor (Creutz and Lagus, 2005), try to maximize the probability of sequences assuming the tokens are generated independently of each other.

3. Background

In this section, we describe how language model based sequence segmentation can be conceptualized via graphs in three parts: (1) constructing a graph where all possible segmentations (i.e., solutions) correspond with paths from a *source* to a *sink* node, (2) equating the shortest path search to the maximum likelihood model, (3) and extending the graph to reflect higher order probability models.

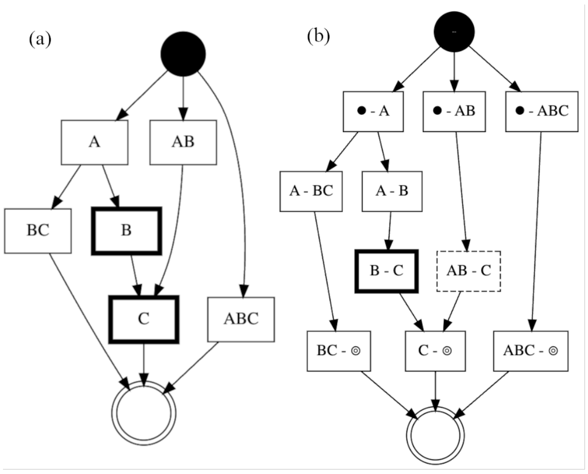

Figure 1: Illustrated unigram (a) and bigram (b) segmentation graphs for the example sequence $\texttt{ABC}$. In bold, we indicate how an example pair of neighboring nodes, namely $\texttt{B}$ and $\texttt{C}$, corresponds with a single node in the bigram graph, $\texttt{B-C}$. Notice how, in the bigram graph, the production of the subsequence $\texttt{C}$ is scored separately in the context of $\texttt{B}$ and $\texttt{AB}$ (dashed box).

3.1. From Sequences to Graphs

Suppose a sentence S of length m is the set of *atoms* a, each being a tuple $(\text{position}, \text{character})$:

$$S = \{a_1, ..., a_m\} = \{(p_1, c_1), ..., (p_n, c_m)\}$$

To segment the sentence means to divide it into n subsets $\pi = \{w_1, ..., w_n\}$, which are (1) pairwise disjoint (non-overlapping), (2) exhaustive (spanning the entire sequence), and (3) subsequences, i.e., it must be possible to arrange the atoms of each w such that the difference between any two successive positions is equal to 1. We will denote by $\mathsf{Subseq}(S)$ the set of *all* candidate subsequences which can be formed from the original sentence.

Consider the example atomic sequence:

$$S = \texttt{ABC} = \{(1, \texttt{A}), (2, \texttt{B}), (3, \texttt{C})\}.$$

To build the *unigram segmentation graph*, we first enumerate all $w \in \mathsf{Subseq}(S)$, namely:

$$\mathsf{Subseq}(S) = \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{AB}, \texttt{BC}, \texttt{ABC}\}.$$

These form the basis for the graph vertices V. We create the edges E by connecting each vertex w_i to an other vertex w_j, if they are adjacent in S, i.e. the maximal atom position in w_i is exactly one less than the minimal position in w_j. The graph is then completed by including a special *begin* and *end*

	Word			*Morfessor*			*BERT*		
	EN	TR	ZH	EN	TR	ZH	EN	TR	ZH
# tokens	61.9M	41.0M	51.6M	69.2M	58.4M	67.4M	64.9M	59.1M	79.7M
# types	262.8K	774.9K	476.4K	33.6K	180.6K	37.3K	24.7K	20.7K	10.5K
tokens/sent.	5.6	4.0	5.1	6.3	5.8	6.6	5.9	5.8	7.9
char./token	3.8	5.7	1.6	3.4	4.0	1.2	3.6	3.9	1.0

Table 1: Corpus statistics. Notice that, using the Huggingface BERT tokenizer, (1) the segmentation was nearly equivalent to character segmentation in case of Chinese (ZH; char./token=1), (2) on average, the tokens were almost one third shorter than words in Turkish (TR), (3) and, in English (EN), the tokens were only marginally shorter than words on average. A similar analysis holds for the case of Morfessor based segmentations, the main difference to the BERT segmentations being that Turkish and Chinese tokens were slightly longer, while English tokens were shorter, on average. The subword vocabularies were roughly an order of magnitude more compact than word vocabularies, the largest ones emerging in case of Turkish.

vertex, v_B and v_E, and (1) connecting the former to all vertices where the minimal position is 1 (thus A, AB, ABC), and (2) connecting vertices where the maximal position is $|S| = 3$ (thus C, BC, ABC) to v_E, the end vertex. See Figure 1a for an illustration.

Solving the segmentation problem now corresponds with finding the best path from v_B to v_E among the set of all such paths – the solution set – which we denote $\mathsf{Paths}(V, E)$.

3.2. Shortest Path and Maximum Likelihood

The data structure is now suitable for the decoding of the most likely sequence of segments according to a probabilistic language model. Interpreting the subsequences associated with the graph nodes as the outcomes of a categorical random variable, which is identically distributed (but not necessarily independent across position), we can assign each edge a weight that is based on the generation probability of the node it points to.

Scoring any particular segmentation, i.e. path $\pi \in \mathsf{Paths}(G)$, thus translates to computing the product of edge weights:

$$L(\pi) = \prod_{w \in \pi} p(w). \quad (1)$$

In practice, we maximize likelihood by minimizing its negative logarithm (NLL):

$$\pi_{\text{best}} = \underset{\pi \in \mathsf{Paths}(V,E)}{\arg\min} -\log(L(\pi)) \quad (2)$$

allowing us to score a candidate path as the sum of log-probabilities, because of this equivalence:

$$\log(\prod_{w \in \pi} p(w)) = \sum_{w \in \pi} \log(p(w)).$$

The problem formulation in terms of NLL is convenient, because conventional pathfinding algorithms are designed with the objective of minimizing the sum of edge weights.

3.3. Higher-Order Graphs

One way to create a higher order graph is by recursively creating a linegraph-like version of its previous-order graph, starting from the unigram version (similarly to how higher-order state-spaces are created in Markov models). Doing so once transforms each pair of adjacent nodes into a new node, now representing a bigram. An illustration of such a bigram graph can be seen in Figure 1b. The final shortest path in such a graph corresponds with a probability model involving a single additional dependency at every position in the sequence:

$$L(\pi) = p(w_1|v_B)...p(w_m|w_{m-1})p(v_E|w_m) \quad (3)$$

where the special vertices v_B and v_E can be interpreted as beginning-, and end-of-sentence tokens. By repeating the procedure, any n-gram graph can be derived.

4. Corpus

In the next sections, we present two experiments, both of which were conducted on the basis of OpenSubtitles [2] (Lison and Tiedemann, 2016), a movie subtitle corpus, which is part of the OPUS corpus (Tiedemann, 2012). We adapted the corpus by taking the overlapping set of documents (movies) between the English, Turkish and (simplified) Chinese subtitles. The resulting intersection was then pre-processed with the objective of keeping the alphabet minimal and language specific: (1) lowercasing, (2) removing punctuation, (3) removing characters that are not from the processed language, (4) and replacing all digit strings with the hashkey (#) character.

Lastly, we divided the corpus into a training and testing portion, using 90% of the subtitle lines for the former and 10% for the latter.

See the corpus statistics in Table 1.

[2] http://www.opensubtitles.org/

5. Experiment 1

The first experiment compares language modeling performance of three (sub-)word segmentation strategies as a function of context size.

5.1. Segmentation Strategies

We selected three competing types of segmentation: word segmentation, Morfessor based subword segmentation, and segmentation produced by popular tokenizers from the Huggingface python library [3].

Word segmentation was obtained by simply taking the tokenized versions of the subtitle corpus. According to Lison and Tiedemann (2016), the English subtitles were tokenized by the Moses toolkit (Koehn et al., 2007) and the Chinese subtitles were tokenized by the KyTea library (Neubig et al., 2011).

Morphologically similar *subword* segmentation was obtained via the Morfessor library (Virpioja et al., 2013). For English and Turkish, we trained the unsupervised baseline model on word counts provided by the latest edition of the MorphoChallenge (Kurimo et al., 2010); for Chinese, we trained on word counts derived from the training split of the subtitle corpus.

BERT tokenizer *subword* segmentations were obtained from the following pre-trained Huggingface models: 'bert-base-uncased' for English, 'dbmdz/bert-base-turkish-uncased' for Turkish, and 'bert-base-chinese' for Chinese. These tokenizers are variants of the BPE (Sennrich et al., 2016) algorithm and are arguably the most widely used text segmenters in industry and academic research related to modern language models.

5.2. Evaluation

For every combination of language and segmentation type, we fitted an *n-gram* count-based language model of order up to 5 on the training split of the dataset, and evaluated it on the testing split.

The results are reported as average values of *bits-per-sentence* (BPS), i.e. the sum of negative log-probabilities of tokens in one line of the test corpus:

$$\mathrm{BPS}(\pi) = -\log(\mathrm{L}(\pi))$$

where $\mathrm{L}(\pi)$ is defined by Equation 1 for unigram models, and Equation 3 for n-gram models where $\mathrm{n} > 1$. We used BPS rather than BPC[4] (bits-per-character), because the former allows for easier comparison between languages and, in this case, only skews the results negligibly since the information content was roughly controlled for by using the same set of movies for each language.

Merely reporting the n-gram order as context size would be misleading, because the size of a particular n-gram with respect to the sentence size depends on the segmentation strategy and language. To account for this variability, we report the context size as the fraction of sentence length (in characters) that the portion of the n-gram used as context amounts to on average. The exact value was computed according to the following formula:

$$(\mathrm{n}-1) * \overline{\mathrm{TL}}/(\overline{\mathrm{SL}} + \mathrm{n} - 1)$$

where n denotes the order of the n-gram, and $\overline{\mathrm{TL}}, \overline{\mathrm{SL}}$ denote the mean token and sentence length in characters. The term $\mathrm{n}-1$ is added to the mean sentence length in the denominator because we pre-pended one single-character padding token for every n-gram order increase above 1 to every sentence.

We used the NLTK (Bird et al., 2009) implementation and the back-off strategy (Katz, 1987) to score unseen words and n-grams: if a particular n-gram does not exist, an $(\mathrm{n}-1)$-gram (containing one less context token) is attempted. If all of the attempts – including the unigram – fail, a logscore derived from frequency 1 is used.

5.3. Results

The results are visualized in Figure 2.

A shared pattern across settings is the reduction of surprisal with higher amount of context.

In the case of English, the differences in scores between the segmentation methods were very small across the entire observed range of context size, and, in contrast to Turkish and Chinese, the word segmentation scored marginally but overall better than the other segmentations.

With Turkish, there was a stronger difference between segmentation methods in terms of surprisal reduction rate. While the word segmentation was still the most optimal at the unigram setting, at around 20% of sentence length used as context, the ranking reversed in favor of the segmentations with shorter tokens: Morfessor and BERT. Between these two, however, the difference was minimal to none.

For Chinese, we observed a similar pattern of ranking reversal at around 20% of context size. The difference to the case of Turkish was mainly an overall faster decline of surprisal values, and an additional difference in rates between Morfessor and BERT segmentations, the latter ranking as most optimal and fastest declining.

[3] https://huggingface.co

[4] $\mathrm{BPC} = \mathrm{BPS}/|\mathrm{S}|$

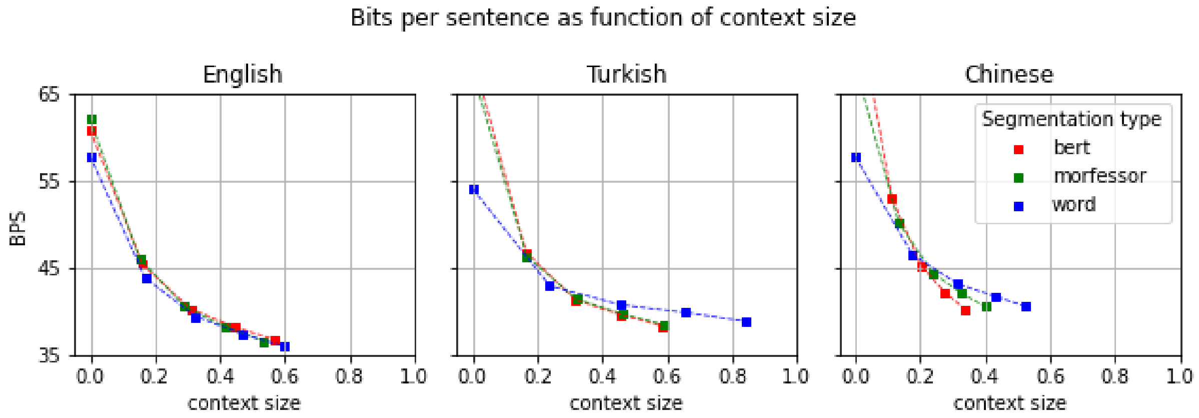

Figure 2: Results of the Experiment 1. The y-axis represents bits-per-sentence on the same scale for each language. The x-axis values correspond with context size measured as proportion of sentence length, 0 implying the unigram language model. The change in model order that modulates this metric (1 to 5, in increments of 1) results in different sentence length proportions due to differently sized tokens: a single Turkish token covers, on average, longer fractions of the sentence than an English or a Chinese token.

5.4. Discussion

Our results indicate that the optimal elementary unit of analysis for language modeling is not universal, but that it depends on the specific characteristics of the language and writing systems.

Previous experiments in language modeling with Chinese text demonstrated better performance for character-based models compared to their word-based counterparts (Li et al., 2019; Mielke et al., 2019), which aligns with our result of word segmentation having higher surprisal rates than character-based segmentation (BERT). Then again, the slightly coarser Morfessor tokens did better than character-based segmentation, indicating that some chunking of Chinese characters might be meaningful.

Similarly, in a study about the impact of Turkish tokenization on language model performance, Toraman et al. (2023) reported that models trained on finer-grained BPE-based segmentations outperform more coarse morphological, word, and character based segmentations. Similarly, the BERT and Morfessor-based segmentation outperformed the word-based segmentation in our experiments.

Previous work comparing English tokenization strategies mainly focused on subword segmentations and recommends using morphologically aligned segmentations over BPE-based techniques (Mielke et al., 2019; Bostrom and Durrett, 2020; Park et al., 2021). However, our results suggest that the English word is not less optimal than other subword units for segmentation. This may have been overlooked in other studies in which the size of context used to predict the next token was not properly controlled for. Word-based segmentation may also be less useful in practice because it requires a larger vocabulary.

Our finding that subword segmentations in Turkish and Chinese benefit from more context is somewhat puzzling, since both Morfessor and BPE discard between-token dependencies during training. In the case of Morfessor, the addition of a prior term regulating vocabulary size in the training procedure could be a contributing factor.

6. Experiment 2

In the second experiment, the goal was to assess the benefits of using a higher-order language model to detect *word* boundaries. To put the effect in perspective, we compare it to a baseline effect of using an increasingly large language sample as the basis for the language model.

To simulate word segmentation, we deleted any within-sentence word boundary notation (i.e. all punctuation and whitespaces) from the test corpus on a per-sentence basis. Every such sentence was then segmented by, first, constructing a segmentation graph (as described in Sections 3.1 and 3.3) and, second, finding the shortest path (defined by Equation 2). The procedure was repeated using uni-, bi-, and trigram language models fitted on word counts derived from increasingly larger samples of sentences, i.e. movie subtitle lines, of the training corpus. The sentences were sampled without replacement and in quantities ranging from 10^2 to 10^7 with integer increments in order of magnitude. The 7th order of magnitude was the full corpus; for all orders of magnitude less than that, we collected 3 differently seeded samples to account for random variation.

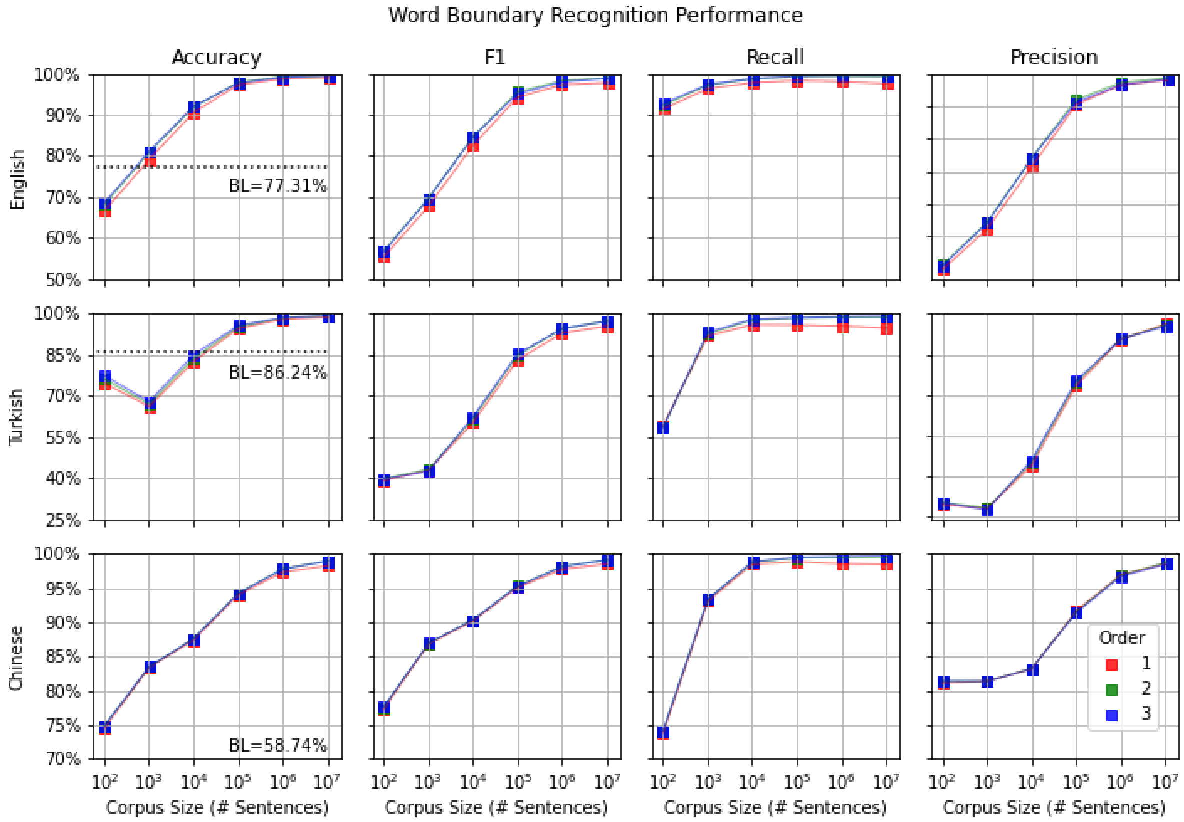

Figure 3: Illustrated results of Experiment 2. The x-axis depicts the corpus size as number of sentences on a log-10 scale. Notice that, while the top value on the y-axis is always 100%, the bottom value is language-specific due to widely different ranges overall. The points in green, corresponding with bigram segmentation models, mostly overlap with the blue (trigram) points.

6.1. Evaluation

The solutions – paths in the segmentation graphs – were converted to sequences of positive and negative labels that correspond with potential word boundaries and indicate whether the position is a boundary or not. The task was evaluated as binary classification, using accuracy, F1-score, recall and precision as performance metrics. The labels were not well balanced: more negative than positive cases are to be expected, but to a different extent for each language. The proportion of cases with the majority label determines our baseline accuracy values.

We evaluated the results on a heldout test-set containing 10000 sentences. The test set differed slightly for each setting of the language model n-gram order: for the unigram model, the sentences were up to 100 characters long, for the bigram model up to 40, and for the trigram model up to 30. We did this because of the different computational demands of the higher-order segmentation. In practice, this means that the unigram models were evaluated on a larger test-set, although only by a little, because the distribution of sentence length was skewed towards values within the 30 character range.

6.2. Results

The results are listed in Table 2. Given the overall trends in Figure 3, we decided to aggregate the order effects separately for small and large models, corresponding with the left and right ranges of corpus size in the figure's plots.

One immediate observation in the Figure 3 is that, in comparison to the baseline effect of sample size, the model order had minimal effect on all performance metrics. The largest, among the small model order effects, was that of bigram vs. unigram model i.t.o. recall. The increment to trigram model, mostly resulted in no further benefits. Overall, the accuracy values closely approached the 100% mark in all three languages when the sample size was the largest. Another general trend was that of rapid recall onset, but lagging rise in precision, which also manifested in F1-scores somewhat lagging behind accuracy scores. With recall, although being high in general, there was a subtle decreasing trend in case of all unigram models.

	English	Turkish	Chinese
Order Effect (models sized 10^2 to 10^4)			
1 to 2	1.53 ± .45	1.22 ± .96	0.21 ± .11
2 to 3	0.31 ± .22	1.15 ± 1.02	0.00 ± .02
Order Effect (models sized 10^5 to 10^7)			
1 to 2	0.53 ± .08	0.54 ± .10	0.41 ± .17
2 to 3	-0.11 ± .06	0.23 ± .19	-0.05 ± .03
Sample Size Effect			
2 to 3	12.55 ± 1.42	-9.06 ± 2.19	8.73 ± 0.78
3 to 4	11.14 ± 0.79	16.70 ± 1.33	4.07 ± 0.53
4 to 5	6.15 ± 0.47	11.48 ± 0.81	6.61 ± 0.55
5 to 6	1.39 ± 0.09	3.15 ± 0.24	3.51 ± 0.14
6 to 7	0.35 ± 0.08	0.77 ± 0.04	1.01 ± 0.14

Table 2: Results of the Experiment 2. The unit value is percentage of accuracy in word boundary classification. The upper portion of the table aggregates the effects of increasing the model order in language models trained on small corpora, the center part on larger corpora. The lower part lists the effects of corpus size i.t.o. increments in order of magnitude.

In English, the increase in accuracy was largely due to the sample size increments from 10^2 to 10^5. From 10^6 on, the accuracy was $> 99\%$. The effect of increasing the model order from 1 to 2 in the smaller models was dwarfed by the sample size effect; with larger models, however, its value of 0.53% was non-negligible compared to the sample size effects $(1.39\%, 0.35\%)$.

In Turkish, we found the singular case of decrease in accuracy due to an increase in sample size, namely from 10^2 to 10^3, matching a dip in precision at this value. The baseline accuracy for Turkish was the highest, and surpassed only by models trained on 10^5 and more sentences. The observations about the effect of model order in English also translate to Turkish.

In Chinese, although the accuracy values grew the slowest, the difference to baseline values was the largest. The pattern of quick onset of recall and lagging precision was also the most marked. The effect of model order was weak with the smaller models, but, with larger models, the increment from 1 to 2 resulted in an accuracy increase of 0.41%, which is non-negligible in comparison to the sample size effect from 10^6 to 10^7 of 1.01%.

6.3. Discussion

The results indicate that, for statistical word segmentation, working with a high quality language sample is important. Segmenting the text with a bigram instead of unigram model can result in further increase in accuracy, although this effect is subtle and only relevant once the language sample is representative enough.

This finding supports the current trend of using unigram-decoder based text tokenizers, which are convenient for their low computational requirements. However, for use-cases where accuracy matters, such as recovering words or morphemes – tokens with precise linguistic definitions –, bigram model based segmentation is recommended. In future work, it would be interesting to explore whether higher-order segmentation aids in, e.g., morphological segmentation or syllabification.

The decline in recall between the unigram and bigram based segmentations is in line with the findings of Goldwater et al. (2009), who connected the independence assumption to undersegmentation. In our findings, larger unigram models did not have problems over-diagnosing boundaries. Although the sensitivity dropped somewhat for unigram models, the higher-order models did not suffer a decline.

7. General Discussion

The two presented experiments explore two different aspects of the role of context in text segmentation. The first experiment examined the difference context makes when evaluating competing segmentation methods. The second experiment looked at the effect of context on statistical word segmentation.

The results suggest that context plays a definitive role in evaluating segmentation methods: the optimal way to encode language is specific to the amount of context used for discovering the regularities in token occurrence. However, we observed, that this is language specific. Surprisingly, our findings also revealed that English word segmentation was on par with the two subword segmentations.

Looking at statistical word segmentation only, the role of context was observable but in small magnitude. While perhaps trivial, this observation is reassuring. It suggests that the inference of distributions governing the sub-lexical regularities (i.e., tokenizers) does not depend on jointly inferring super-lexical regularities, which would severely complicate the procedure. It further implies that, to the extent that written text mirrors properties of spoken language, this offers an explanation on how children can learn to discern words while being unaware of higher-level dependencies between them due to, e.g., syntax or semantics, which they learn at later stages of development.

8. Conclusion

The role of preceding text in tokenization was manifested in two ways. When comparing the difficulty in modeling differently tokenized corpora, our results indicate that the assessment may fully reverse when context is involved compared to when it is absent. In light of a word segmentation experiment, the role was more subtle: word boundaries were only marginally more accurately recognized when using context-sensitive, rather than context free, methods to score the hypotheses.

9. Acknowledgements

This research was funded by ITK as part of the SmartDATA (10028312) project as part of the KPN Responsible AI lab. We also thank the reviewers for their insightful comments.

10. Limitations

In this work, we only computed language modeling surprisal values on the basis of count-based language models. The extent to which these results generalize to other types of language models (e.g. neural network based) is unclear.

11. Bibliographical References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. 2016. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182, New York, New York, USA. PMLR.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

M A Anusuya and S K Katti. 2009. Speech recognition by machine: A review. *IJCSIS) International Journal of Computer Science and Information Security*, 6.

Eleanor Olds Batchelder. 2002. Bootstrapping the lexicon: A computational model of infant speech segmentation. *Cognition*, 83:167–206.

Yves Bestgen. 2006. Improving text segmentation using latent semantic analysis: A reanalysis of choi, wiemer-hastings, and moore (2001). *Computational Linguistics*, 32:5–12.

Narin Bi and Nguonly Taing. 2014a. Khmer word segmentation based on bi-directional maximal matching for plaintext and microsoft word document. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–9. IEEE.

Narin Bi and Nguonly Taing. 2014b. Khmer word segmentation based on bi-directional maximal matching for plaintext and microsoft word document. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–9.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Michael R Brent. 1999. Speech segmentation and word discovery: A computational perspective. *Trends in Cognitive Sciences*, 3(8):294–301.

Vichet Chea, Ye Kyaw Thu, Chenchen Ding, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2015. Khmer word segmentation using conditional random fields. *Khmer Natural Language Processing*, pages 62–69.

Aitao Chen. 2003. Chinese word segmentation using minimal linguistic knowledge. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 148–151, Sapporo, Japan. Association for Computational Linguistics.

Aitao Chen, Jianzhang He, Liangjie Xu, Fredric C. Gey, and Jason Meggs. 1997. Chinese text retrieval without using a dictionary. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '97, page 42–49, New York, NY, USA. Association for Computing Machinery.

Keh-Jiann Chen and Shing-Huan Liu. 1992. Word identification for mandarin chinese sentences.

Grzegorz Chrupała. 2023. Putting natural in natural language processing. *arXiv preprint arXiv:2305.04572*.

Ronald A Cole, Jola Jakimik, William E Cooper, and Joia Jakimik. 1980. Segmenting speech into words. *J. Acoust. Soc. Am*, 67:1323–1332.

Mathias Creutz and Krista Lagus. 2005. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology Helsinki.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4.

Matthew H. Davis, William D. Marslen-Wilson, and M. Gareth Gaskell. 2002. Leading up the lexical garden path: Segmentation and ambiguity in spoken word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 28:218–244.

Ke Deng, Peter K. Bol, Kate J. Li, and Jun S. Liu. 2016. On the unsupervised analysis of domain-specific chinese texts. *Proceedings of the National Academy of Sciences of the United States of America*, 113:6154–6159.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Miguel Domingo, Mercedes Garcıa-Martınez, Alexandre Helle, Francisco Casacuberta, and Manuel Herranz. 2019. How much does tokenization affect neural machine translation?

Nadir Durrani and Sarmad Hussain. 2010. Urdu word segmentation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 528–536.

Schubert Foo and Hui Li. 2004. Chinese word segmentation and its effect on information retrieval. *Information Processing and Management*, 40(1):161–190.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Matthias Gallé. 2019. Investigating the effectiveness of BPE: The power of shorter sequences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1375–1381, Hong Kong, China. Association for Computational Linguistics.

Xianping Ge, Wanda Pratt, and Padhraic Smyth. 1999. Discovering chinese words from unsegmented text. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999*, pages 271–272.

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

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.

Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep speech: Scaling up end-to-end speech recognition. *CoRR*, abs/1412.5567.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark. Association for Computational Linguistics.

Chea Sok Huor, Top Rithy, Ros Pich Hemy, Vann Navy, Chin Chanthirith, and Chhoeun Tola. 2004a. Word bigram vs orthographic syllable bigram in khmer word segmentation. *PAN Localization Working Papers*, 2007.

Chea Sok Huor, Top Rithy, Ros Pich Hemy, Vann Navy, Chin Chanthirith, and Chhoeun Tola. 2004b. Word bigram vs orthographic syllable bigram in khmer word segmentation. *PAN Localization Working Papers*, 2007.

Peter W. Jusczyk, Anne Cutler, and Nancy J. Redanz. 1993. Infants' preference for the predominant stress patterns of english words. *Child Development*, 64:675.

S. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401.

Dennis H. Klatt and Kenneth N. Stevens. 1973. On the automatic recognition of continuous speech: Implications from a spectrogram-reading experiment. *IEEE Transactions on Audio and Electroacoustics*, 21:210–217.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Hideki Kozima. 1996. Text segmentation based on similarity between words.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. `https:`

//taku910.github.io/mecab/. Accessed: 2023-12-04.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Mikko Kurimo, Sami Virpioja, Ville T. Turunen, Graeme W. Blackwood, and William Byrne. 2010. Overview and results of Morpho Challenge 2009. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 – October 2, 2009, Revised Selected Papers*, volume 6241 of *Lecture Notes in Computer Science*, pages 578–597. Springer Berlin / Heidelberg.

Mun-Kew Leong and Hong Zhou. 1997. Preliminary qualitative analysis of segmented vs bigram indexing in chinese.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of chinese representations?

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Józef Maciuszek. 2018. Lexical access in the processing of word boundary ambiguity. *Social Psychological Bulletin*, 13.

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara. 2000. Morphological analysis system chasen version 2.2. 1 manual. *Nara Institute of Science and Technology*.

Sven L. Mattys, Peter W. Jusczyk, Paul A. Luce, and James L. Morgan. 1999. Phonotactic and prosodic effects on word segmentation in infants. *Cognitive Psychology*, 38:465–494.

Surapant Meknavin, Paisarn Charoenpornsawat, and Boonserm Kijsirikul. 1997. Feature-based thai word segmentation.

S Meknawin. 1995. Towards 99.99% accuracy of thai word segmentation. In *Oral Presentation at the Symposium on Natural Language Processing in Thailand*, volume 95.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp.

Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What kind of language is hard to language-model? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989, Florence, Italy. Association for Computational Linguistics.

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. 2012. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 8(67).

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA. Association for Computational Linguistics.

Jian Yun Nie, Martin Brisebois, and Xiaobo Ren. 1996. On chinese text retrieval. *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 225–234.

Shu Okabe, Laurent Besacier, and François Yvon. 2022. Weakly supervised word segmentation for computational language documentation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7385–7398, Dublin, Ireland. Association for Computational Linguistics.

Hyunji Hayley Park, Katherine J Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: a multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Lawrence Phillips and Lisa Pearl. 2014. Bayesian inference as a cross-linguistic word segmentation strategy: Always learning useful things. In *Proceedings of the 5th Workshop on Cognitive*

Aspects of Computational Language Learning (CogACLL), pages 9–13, Gothenburg, Sweden. Association for Computational Linguistics.

Yuen Poowarawan. 1986. Dictionary-based thai syllable separation. In *Proc. Ninth Electronics Engineering Conference (EECON-86), Thailand*, pages 409–418.

A Pornprasertkul. 1994. *Thai syntactic analysis*. Ph.D. thesis, Ph. D Thesis, Asian Institute of Technology.

S Rarunrom. 1991. Dictionary-based thai word separation. *Senior Project Report*.

D. Raj Reddy. 1976. Speech recognition by machine: A review. *Proceedings of the IEEE*, 64:501–531.

Jenny R Saffran, Richard N Aslin, and Elissa L Newport. 1996a. Statistical learning by 8-month-old infants. *New Series*, 274:1926–1928.

Jenny R. Saffran, Elissa L. Newport, and Richard N. Aslin. 1996b. Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 35:606–621.

Masahiko Sato. 1999. Kakasi - kanji kana simple inverter. `http://kakasi.namazu.org`. Accessed: 2023-12-04.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Richard Sproat and Thomas Emerson. 2003. The first international Chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143, Sapporo, Japan. Association for Computational Linguistics.

Richard Sproat and Chilin Shih. 1990. A statistical method for finding word boundaries in chinese text. *Computer Processing of Chinese and Oriental Languages*, 4:336–351.

Maosong Sun, Dayang Shen, and Benjamin K. Tsou. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 1265–1271, Montreal, Quebec, Canada. Association for Computational Linguistics.

Zhiqing Sun and Zhi Hong Deng. 2018. Unsupervised neural word segmentation for chinese via segmental language modeling. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 4915–4920.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Channa Van and Wataru Kameyama. 2013. Khmer word segmentation and out-of-vocabulary words detection using collocation measurement of repeated characters subsequences. *GITS/GITI Research Bulletin*, 2012-2013:21–31.

Anand Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27:351–372.

Sornlertlamvanich Virach. 1993. Word segmentation for thai in machine translation system. *Machine translation*.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical report, Aalto University.

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

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*.

12. Language Resource References

Lison, Pierre and Tiedemann, Jörg. 2016. *Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles*. European Language Resources Association (ELRA).

Tiedemann, Jörg. 2012. *Parallel Data, Tools and Interfaces in OPUS*. European Language Resources Association (ELRA).