DP-Parse: Finding Word Boundaries from Raw Speech with an Instance Lexicon

Robin Algayres[1,3,5] **Tristan Ricoul**[3] **Julien Karadayi**[3] **Hugo Laurençon**[3]
Salah Zaiem[3] **Abdelrahman Mohamed**[2] **Benoît Sagot**[4] **Emmanuel Dupoux**[1,3,4,5]

Meta AI Research, France[1], Meta AI Research, USA[2], ENS/PSL, Paris, France[3],
EHESS, Paris, France[4], Inria, Paris, France[5],
{robin.algayres, benoit.sagot}@inria.fr
dpx@fb.com

Abstract

Finding word boundaries in continuous speech is challenging as there is little or no equivalent of a 'space' delimiter between words. Popular Bayesian non-parametric models for text segmentation (Goldwater et al., 2006, 2009) use a Dirichlet process to jointly segment sentences and build a lexicon of word types. We introduce DP-Parse, which uses similar principles but only relies on an *instance lexicon* of word tokens, avoiding the clustering errors that arise with a lexicon of word types. On the Zero Resource Speech Benchmark 2017, our model sets a new speech segmentation state-of-the-art in 5 languages. The algorithm monotonically improves with better input representations, achieving yet higher scores when fed with weakly supervised inputs. Despite lacking a type lexicon, DP-Parse can be pipelined to a language model and learn semantic and syntactic representations as assessed by a new spoken word embedding benchmark. [1]

1 Introduction

One of the first tasks that infants face is to learn the words of their native language(s). To do so, they must solve a word segmentation problem, since words are rarely uttered in isolation but come up in multi-word utterances (Brent and Cartwright, 1996). Segmenting utterances into word or subword units is also an important step in NLP applications, where an input string of orthographic symbols is *tokenized* into words or sentence piece before being fed to a language model. While many writing systems use white spaces that make (basic) tokenization a relatively simple task, others do not (e.g., Chinese) and turn tokenization into a challenging machine learning problem (Li and Yuan, 1998). A similar situation arises for 'textless' language models based on units derived from raw audio (Lakhotia et al., 2021), which, like infants, do not have access to isolated words or word-separating symbols.

The most successful approach to unsupervised word segmentation for text inputs is based on *non-parametric Bayesian models* (Goldwater et al., 2006, 2009; Kawakami et al., 2019; Kamper et al., 2017b; Berg-Kirkpatrick et al., 2010; Eskander et al., 2016; Johnson et al., 2007; Godard et al., 2018) that jointly segment utterances and build a lexicon of frequent word forms using a Dirichlet process. The intuition is that frequent word forms function as anchors in a sentence, enabling the segmentation of novel words (like 'dax' in 'did you see the dax in the street?'). Such models are tested on *phonemized texts* obtained by converting text into a stream of phonemes after removing spaces and punctuation marks. Even though phonemized text may seem a reasonable approximation of continuous speech, such models have given disappointing results when applied directly to speech inputs. Since the direct comparison between speech-based and text-based models was introduced in 2014 by Ludusan et al. (2014), the gap of performance between these two types of inputs as documented in three iterations of the Zero Resource Speech Challenge focused on word segmentation has remained large (Versteegh et al., 2016; Dunbar et al., 2020). We attribute these difficulties to two major challenges posed by speech compared to text inputs: acoustic variability and temporal granularity, which we address in our contribution.

The first and most important challenge is *acoustic variability*. In text, all tokens of a word are

[1]Our open-source implementation of DP-Parse: https://gitlab.cognitive-ml.fr/ralgayres/dp-parse.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 1051–1065, 2022. https://doi.org/10.1162/tacl_a_00505
Action Editor: Mauro Cettolo. Submission batch: 4/2022; Revision batch: 6/2022; Published 9/2022.

represented the same. In speech, each token of a word is different, depending on background noise, intonation, speaker voice, speech rate, and so forth. Text-inspired speech segmentation algorithms (Kamper et al., 2017b) apply a clustering step to word tokens in order to get back to word types. We believe that this step is unnecessary and is responsible for the low performance of existing systems, as errors in the clustering step are not recoverable and negatively impact word segmentation. We propose an algorithm that segments speech utterances based on an *instance lexicon* of word tokens, instead of a lexicon of word types. Each speech segment instance of the training set is represented as a distinct memory trace, and these traces are used to estimate word form frequency using a k-NN algorithm without having to refer to a discrete word type but still applying the same Dirichlet process logic. As for the representation of these word tokens, we follow recent approaches that use fixed length *Speech Sequence Embeddings (SSE)* (Thual et al., 2018; Kamper et al., 2017b). We use state-of-the-art SSEs from Algayres et al. (2022) that have either been trained in a self-supervised fashion or with weak labels, in order to assess how the segmentation model behaves with inputs of increasing quality.

The second challenge is related to the fact that the speech waveform being *continuous* in time, the number of possible segmentation points grows very large, making the optimization of segmentation of large speech corpora using Bayesian techniques *intractable*. Some models reduce the number of segmentation points using phoneme boundaries (Lee and Glass, 2012; Bhati et al., 2021), syllable boundaries (Kamper et al., 2017b), or a constant discretization of the time axis into speech frames, subject to the following trade-off: Too coarse frames may miss some short word boundaries, too fine-grained ones may render segmentation intractable on large corpora because of the quadratic increase in number of segmentations with frame rate. Here we choose a compromise with 40ms frames, corresponding to half of the mean duration of a phoneme, yielding a 4x theoretical slowdown compared to phoneme-based transcripts. In addition, we introduce an *accelerated version* of a Dirichlet process segmentation algorithm that replaces Gibbs sampling of each boundary by sampling entire *parse trees* using Dynamic Programming Beam Search, and updating the lexicon in *batches* instead of continuously. This achieves a 10-times speed-up compared to Johnson et al. (2007)'s implementation, with similar or superior performances.

Because of our 'no clustering' approach, we cannot evaluate our model using the word type-based metrics of the Zero Resource speech segmentation challenge (Versteegh et al., 2016; Dunbar et al., 2017). Here, we use the two classes of metrics: boundary-related metrics (Token and Boundary F-score) that do not refer to types, and high-level word embedding metrics. The idea is use a downstream language model to learn high level word embeddings and evaluate them on semantic and Part-Of-Speech (POS) dimensions. We rely on a semantic similarity task from Dunbar et al. (2020) and introduce two new metrics.

The combination of our contributions yields an overall system, DP-Parse, that sets a new state-of-the-art on speech segmentation for five speech corpora by a large margin. By doing various ablation and anti-ablation studies (replacing unsupervised components by weakly supervised ones) we pinpoint the components of the system where there is still margin for improvement. In particular, we show that using better embeddings obtained with weak supervision, it is possible to double the segmentation F-score and substantially improve our new semantic and POS scores.

2 Related Work

2.1 Speech Sequence Embeddings

Speech Sequence Embedding (SSE) models take as input a piece of speech signal of any length and outputs a fixed-size vector. The main objective of these models is to represent the phonetic content of a given speech segment. A naive SSE model would be to extract frame-level features of a speech sequence, using for instance Wav2vec2.0 or HuBERT (Baevski et al., 2020; Hsu et al., 2021), and mean-pool the frames along the time axis. A more subtle approach is to train a self-supervised systems on positive pairs of speech sequences. The model learns speech sequence representation thanks to contrastive loss functions (e.g., the infoNCE loss as in Algayres et al. [2022]; Jacobs et al. [2021] or the Siamese loss as in Settle and Livescu [2016]; Riad et al. [2018]) or reconstructive losses (e.g., *Correspondance Auto-Encoder* from Kamper [2018] where the first

element of the pair is compressed and decoded into the second element). Recently, a growing body of work leverage multilingual labeled dataset to build SSEs that can, to an extent, generalize to unseen languages (Hu et al., 2020a,b; Jacobs et al., 2021). In this work, we use a state-of-the-art self-supervised system SSE model from Algayres et al. (2022).

2.2 Speech Segmentation

Three main class of models have been proposed to solve speech segmentation: *matching-first* models (Park and Glass, 2007; Jansen and Van Durme, 2011; Hurtad, 2017) attempt to find high quality pairs of identical segments and cluster them based on similarity, thereby building a lexicon of word types that may not cover the entire corpus (Bhati et al., 2020; Räsänen et al., 2015). *Segmentation-first* models try to exhaustively parse a sentence (Lee et al., 2015; Kamper et al., 2017a,b; Kamper and van Niekerk, 2020), while jointly learning a lexicon (and therefore also involving matching and clustering). *Segmentation-only* models discover directly the likely word boundaries in running speech using prediction error (Bhati et al., 2021; Cuervo et al., 2021), without relying on any lexicon of word types. Our approach is a new hybrid of the last two lines of research: Like a segmentation-first model, we jointly model lexicon and segmentation, but our lexicon is not a lexicon of types, thereby escaping matching and clustering errors, like in segmentation-only models. This model shows good performances across different languages as measured by the Mean Average Precision on pre-segmented spoken words.

2.3 Evaluating Speech Segmentation

Across the different word segmentation studies, the evaluation has been done according to two general classes of metrics: matching and clustering metrics for word embeddings (MAP, NED, grouping F-score, Type F-score), and segmentation metrics for boundaries (Token and Boundary F-scores) (Carlin et al., 2011; Ludusan et al., 2014). All of these metrics presuppose that the optimal segmentation strategy is aligned with the text-based gold standard (based on spaces and punctuation). Yet, it is entirely possible that such segmentation is not optimal for downstream applications, as witnessed by the fact that most current language models use subword units like BPEs or sentence pieces (Devlin et al., 2018). Therefore, we propose a third class of segmentation metrics based on downstream language modeling tasks. Chung and Glass (2018) showed that gold word boundaries provide sufficient information to learn good quality semantic embeddings from raw audio using a skipgram or word2vec objective, as assessed by semantic similarity benchmarks adapted to speech inputs. These datasets were also used to compile the sSIMI benchmark in the Zero Speech Challenge 2021 (Nguyen et al., 2020), but turn out to give very low and noisy results (at least for small training sets). Here, we introduce two new metrics evaluating semantic and grammatical similarity that we show to be more stable.

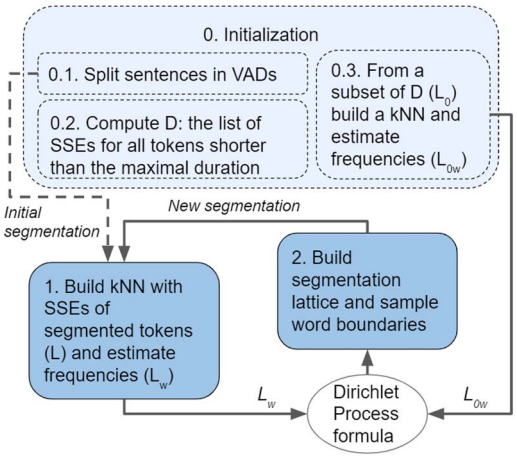

Figure 1: The core components of DP-Parse. The algorithm is a loop between two main steps (Step 1. and 2.) in the spirit of the Expectation Maximization algorithm. Given an initial coarse word segmentation, we estimate L_w and L_{0_w} (i.e., the Dirichlet Process parameters) with k-NN density estimation over SSEs. The estimated parameters are used to derive a new word segmentation, which in turn serves to re-estimate L_w and L_{0_w}.

3 Method: Instance-based Dirichlet Process Parsing

This section describes the building blocks and notations used by DP-Parse as depicted in Figure 1 and Table 1, respectively. At a high level, DP-Parse is a new adaptation of Goldwater's Dirichlet process algorithm (Goldwater et al., 2009) for speech inputs. At the heart of Goldwater's algorithm is a hierarchical Bayesian model that generates a joint distribution of a lexicon of words (a word is a sequence of phonemes) and a corpus (a sequence of words). The associated learning

u	a unit u is a block of 40 ms of speech signal
$w = (u_0, \ldots, u_w)$	a segment w is a sequence of units
$E_w \in \mathbb{R}^p$	a fixed length embedding of a segment w computed by a SSE model
$C = \{v^0, \ldots, v^N\}$	a corpus C is a set of utterances (segments without silence found by VAD)
$C_{seg} = \{(w_1^0, \ldots, w_{p_0}^0), ...\}$	a segmented corpus C_{seg} is a set of sequences of segments
$D = \{w_0, ...w_P\}$	the list of all possible subsegments in C that are possible words (between 40ms and 800ms)
$L_0 = \{E_{w_0}, \ldots, E_{w_P}\}$	a kNN index of embeddings from D
$L = \{E_{w_0}, \ldots, E_{w_n}\}$	a kNN index of all embeddings from the segments in C_{seg}
L_{0_w}	the frequency of E_w in L_0
L_w	the frequency of E_w in L

Table 1: Notations used in Section 3.

algorithm optimizes the likelihood of an observed corpus as a function of the (unobserved) lexicon and corpus segmentation. It does so by alternating between two EM steps: (Step 1) estimating the most probable lexicon given a segmentation, and (Step 2) sampling among the most probable segmentation given the lexicon. More precisely, at each Step 1, the algorithm estimates the probability of a segmentation from two sets of numbers: the probability $P_W^0(w)$ that a given speech segment w is a word, and L_w the frequency of a given word w in the (segmented) input corpus. They are estimated by building tables of counts of phoneme-n-grams and words given a segmentation, respectively. At Step 2, these numbers are combined using Equation 3 (Section 3.3) and a new segmentation is sampled. Our adjustments to the original algorithm are the following:

- Section 3.1: Phonemes are replaced by 40ms speech frames (from a pretrained self-supervised learning algorithm), words are replaced by fixed-length *SSEs* (using a pretrained self-supervised embedder).

- Section 3.2: The tables of counts for computing $P_W^0(w)$ and L_w are replaced by k-Nearest-Neighbors (k-NN) indexes of embeddings (L_0, and L, respectively) which can be viewed as instance lexicons of all possible n-frames and of segmented 'words', respectively. The count estimates are obtained using Gaussian kernel *density estimation* over k-NN.

- Section 3.3: We make a small adaptation to the*Dirichlet process formula*.

- Section 3.4: Instead of alternating between the two steps at each boundary as in the original Gibbs sampling algorithm, which is very time-consuming, we sample segmentations over entire utterances using a *segment lattice*, and update the lexicon over a batch of utterances.

- Section 3.5: We *initialize* the system by selecting as an initial lexicon a list of short enough utterances, and precompute all the constant values (L_0 and $P_W^0(w)$) to optimize for speed.

So modified, the algorithm is no longer dependent on input type and can work either with discrete representations (text, represented as sequences of 1-hot vectors of phonemes) or continuous ones (speech, represented as embeddings).

3.1 Speech Sequence Embeddings

We represent speech as 20ms frames obtained by selecting the 8th layer of a pretrained Wav2vec2.0 Base system from Baevski et al. (2020). Each two successive frames are tied together so that a speech sentence is a series of 40ms speech blocks. To represent a speech segment, we use the SSE model from Algayres et al. (2022): a self-supervised system trained with contrastive learning where positive pairs are obtained by data-augmentation of the speech signal. This model shows good performance across different languages as measured by the Mean Average Precision on pre-segmented spoken words. The SSE model takes as input the pre-extracted Wav2vec2.0 frames and applies a single 1-D convolution layer with GLU (kernel size: 4, number of channels: 512, stride: 1) and a single transformer

layer (attention heads: 4, size of attention matrices: 512 neurons, and feed forward layer: 2048 neurons). A final max-pooling layer along the time axis is applied to obtain a fixed-size vector. To save computation, we reduce the dimensionality of Algayres et al.'s (2022) SSE model from 512 to 64 with a PCA trained on random speech segments extracted from the corpus at work.

In addition to their unsupervised SSE model, Algayres et al. (2022) provide a weakly supervised version. They trained it with the same contrastive loss (infoNCE) using positive pairs obtained with a time-aligned transcription of the speech signal. The weakly supervised version will be used as a topline model, as it scores much higher on Mean Average Precision than the self-supervised one.

3.2 Density Estimation from Gaussian-smoothed k-Nearest-Neighbors

In text, the exact frequency of a string of letters can be computed by counting how often it occurs in the corpus. In speech, due to acoustic variability, most of the counts would be 1, and clustering methods cause too many errors to obtain reliable count estimates (more details in Appendix A.2.1). Here, we follow the method in Algayres et al. (2020) using Gaussian filtering over k-NNs. Let us assume that all speech segments w of a corpus are represented as embeddings E_w and stored in a k-NN index L_0. To compute L_{0_w} the frequency of w in L_0, we extract the k nearest-neighbors of E_w: $(E_1, \ldots, E_k)$[2] and sum over them, weighted by a decreasing function of their distance to w. Intuitively, embeddings close to E_w are more likely to be instances of the same segments than distant ones. For weighting, we use a Gaussian kernel centred on E_w as in the following equation:

$$L_{0_w} \approx F(w) = \sum_{j=1}^{k} \exp^{-\beta \|E_w - E_j\|_2^2} \quad (1)$$

F is the Parzen-Rosenblatt window method for density estimation (Parzen, 1962) and returns a continuous value between 0 and k, estimating L_{0_w}. It rests on two hyper-parameters: k, which is set to 1000, and β, the standard deviation of the Gaussian kernel. To set β, we follow the observation from Algayres et al. (2022): Around 50% of the segments in the development dataset of the Zerospeech Challenge 2017 appear to have a frequency of one. Therefore, we set β during runtime so that 50% of the L_{0_w} calculated get a frequency of one (i.e., an F value below a small $\epsilon > 0$). We did not change these hyper-parameters for any of the tested languages.

[2]We remove from the list the embeddings corresponding to overlapping segments.

3.3 Dirichlet Process Formula

This section explains how Goldwater et al. (2009) formulate the Dirichlet Process and our modification. Let v' be a non silent section from a speech corpus C found by a Voice Activity Detection (VAD). A segmentation of v' is written v'_{seg} and is composed of a series of segment $(w_1, \ldots, w_l)$. The probability of v'_{seg} is, under the unigram hypothesis, the product of the probability of each segment to be a real spoken word:

$$P_S(s'_{seg}) = \prod_{i=0}^{l} P_W(w_i) \quad (2)$$

To model the probability of a segment w to be a real word, Goldwater et al. (2009) use the following formulation of the Dirichlet Process:

$$P_W(w|C, C_{seg}) = \frac{L_w}{\#L + \alpha_0} + \frac{\alpha_0 P_W^0(w)}{\#L + \alpha_0} \quad (3)$$

The first term of Equation 3 accounts for how often the token w has been segmented in C_{seg}. Its numerator L_w is the count estimate of w in L. The second part of Equation 3 is the intrinsic probability of w to be a word regardless of its appearance in C_{seg}. This intrinsic probability is called the base distribution P_W^0 and is controlled by the concentration parameter α_0.

To find a formulation for P_W^0, the intuition from Goldwater et al. (2009) is simple: frequently appearing tokens are more likely to be true words than rare ones. Yet, their formulation of P_W^0 cannot be easily adapted to the segmentation of speech data. Our contribution here is to propose a new formula for P_W^0 that follows the same intuition:

$$P_W^0(w) = \frac{L_{0_w}}{\#L_0} \quad (4)$$

where L_{0_w} is the count estimate of w in L_0 (the lexicon of all possible segments in C). $\#L_0$ being the total number of segments in L_0, P_W^0 is then the probability to find w in among the tokens in C.

3.4 Segmentation Lattice

In Goldwater et al. (2009), the learning algorithm uses Gibbs sampling, where word boundaries are sampled one at a time, requiring all parameters of the Dirichlet Process model to be recomputed after each sample. On text data, it is not a problem, as updating the model's parameters (L_w and L_{0_w}) is fast: No k-NN search is needed, parameters are computed exactly by matching and counting strings. It becomes a bottleneck for speech segmentation where the parameters are computed with density estimation and k-NN search. One way to alleviate this problem is to incorporate the dynamic programming version of the Gibbs sampler from Mochihashi et al. (2009). Another way, as in this paper, is to build a segmentation lattice over each utterance and sample among the n-best segmentations.

Here, we assume that the corpus C has been pre-segmented in a set of utterances using a VAD algorithm. After each Step 1 from Figure 1 we can compute the log probability, $\log(P_W(w))$ from Equation 3, that each token in an utterance is a real word. Yet, instead of directly using this probability, we introduce a per-token penalty score that favors short tokens:

$$q(x) = \left(\frac{x-1}{\delta}\right)^{\gamma} \quad (5)$$

This penalty is added to the Dirichlet process log probability as follows:

$$S(w) = \log(P_W(w) + \epsilon) + q(len(w)) \quad (6)$$

where ϵ is a very small number to handle cases where $P_W(w) = 0$, $len(w)$ is the number of 40ms speech frames in w.

For each utterance in C, we create a segmentation lattice that provides a compact view for all possible segmentation paths. A segmentation path is a sequence of consecutive segmentation arcs that covers the full utterance. Each arc starts and finishes in-between the units and is bounded within a minimal and maximal length. An example of a segmentation lattice can be found in Figure 2. Each segmentation arc is associated with its S score from Equation 6. For each utterance, the n-best segmentations are computed with Dynamic Programming Beam Search, and we sample from a softmax of their total S scores. One advantage of this procedure is that it is possible to parallelize utterance segmentation across large batches and only update the lexicon L after each batch. In our experiments, we take the entire corpus C to be a single batch.

Minimum segment length	$40ms$
Maximum segment length	$800ms$
No. of segments in L_0	1M
Concentration parameter α_0 (eq. 3)	100
No. of neighbors in k-NN (eq. 1)	100
Lattice beam size	10
Duration penalty q (eq. 5)	$\gamma = 1.8$
	$\delta = 2$ (text)
	$\delta = 4$ (speech)

Table 2: DP-Parse hyper-parameters.

3.5 Initializing DP-Parse

We create a corpus C as a collection of utterances by applying the pyannote VAD (Bredin et al., 2019) with a threshold of 200 ms. As shown in Figure 1, initialization (Step 0) contains several sub-steps. The first (0.1) is to provide an initial segmentation of C, using a simple heuristic: All sentences shorter than 800ms are treated as word tokens, the other ones are discarded. The intuition is that short sentences could be words in isolation, which provides the seed for an initial lexicon.

The second sub-step (0.2) is to create the list D composed of the embeddings of all possible speech segments in C that are possible words, which we set to be anything between 40ms and 800ms (by 40ms increments). To embed a speech segment, we use the SSE model in Section 3.1.

The third sub-step (0.3) consists in the construction of L_0: a k-NN index of all embedded segments in D. In practice, we found that randomly subsampling D to one million embedded segments worked well (see grid-search in Appendix A.2.2), and we precompute L_{0_w} as explained in Section 3.2.

4 Evaluation Metrics and Settings

4.1 Boundary Level Segmentation Metrics

In the Zerospeech Challenge 2017 (Dunbar et al., 2017), two metrics measure how well discovered boundaries match with the gold word boundaries. These metrics, the Boundary and Token F-score, are obtained by comparing the discovered sets of tokens to the gold one obtained by force-aligning spoken sentences with their transcription. The evaluation is done in phoneme space and each discovered boundary is mapped to a real phoneme boundary: If a boundary overlaps a phoneme by

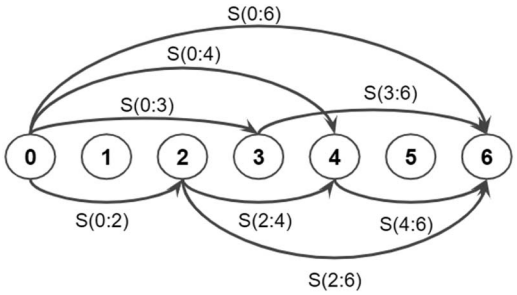

Figure 2: An example of a segmentation lattice for a small utterance of 6 units, with a constraint on word tokens to be bounded between 2 and 6 units. A segmentation path is a sequence of segmentation arcs that covers the whole utterance. Each arc starts and ends in-between units and is associated with its score S from Equation 6.

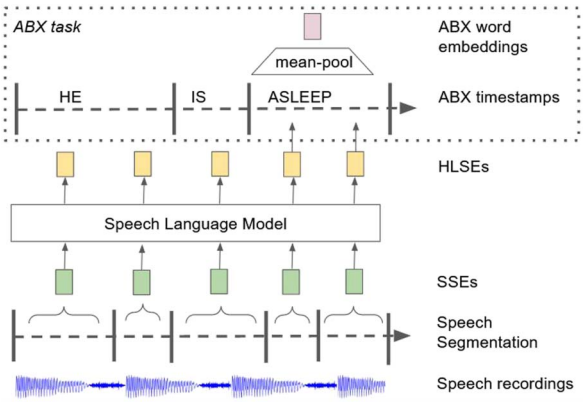

Figure 3: Method for deriving HLSEs for semantic and POS evaluations. Speech is segmented and converted into SSEs before being used to train a Speech LM. At test time, the HLSEs that overlap with the ABX timestamps by more than 40ms are mean-pooled to form the ABX word embeddings (here the word 'asleep').

more than 30ms or more than half of the phoneme length, the boundary is placed after that phoneme (otherwise it is placed before).

4.2 Semantic and POS Embedding Metrics

The idea here is to evaluate segmentation through its effect on a downstream language model. The evaluation is less direct than the segmentation metrics above, as it assumes that the segmentation is used to turn a spoken utterance into a series of fixed size SSEs, themselves used to train a continuous-input Speech Language Model (LM). The metrics will therefore reflect each of these components (segmentation, SSE, Speech LM), but by keeping the SSE and Speech LM components constants, we can hope to study systematically the effect of the segmentation component. Our evaluation focuses on word-level representations, which we call here High-Level Speech Embeddings (HLSE). They are the speech equivalent of Word2Vec representations (Mikolov et al., 2013), yet coming from an inexact segmentation. Here, we assume that a Speech LM has been trained on SSE inputs on a given dataset with a given segmentation and can be used at test time to provide embeddings associated to a set of test words. As our segmentation models work on whole utterances, we present the test words in continuous utterances and then mean pool the HLSEs from the Speech LM that overlap with that word to obtain a single vector. An overview of the building of generic HLSEs and their evaluation is shown in Figure 3.

We introduce two zero-shot tasks to evaluate HLSEs that do not require training a further classifier, and are purely based on distances between these embeddings. They are based on machine-ABX, a type of distance-based metric introduced in Schatz et al. (2013) which computes a discrimination score over sets of (A, B, X) triplets. For each triplet, A and X belong to the same category and B plays the role of a distractor. The task for the model is to find the distractor in each triplet based on the cosine distance between the HLSEs of A, B, and X. The first task (ABX_{sem}), have A and X to be synonyms whereas B is semantically unrelated to neither A or X. The second task, ABX_{POS}, have A and X share the same POS-tags whereas B has a different one. The triplets are all extracted from the Librispeech corpus training set. See Appendix A.1 for details on the construction of the triplets.

Another task, sSIMI from Nguyen et al. (2020), also evaluates high-level representations of spoken words in a zero-shot and distance-based fashion. Yet, sSIMI differs from our ABX tasks in two important ways. First, sSIMI presents test words as pre-segmented chunk of speech without the context of the original sentence to help the Speech LM. Second, the task for the Speech LM is to predict a semantic similarity score given by human annotators to pairs of words. The Speech LM encodes the pre-segmented spoken words into HLSEs and the distance between the HLSEs is correlated with the human judgements. The correlation coefficient r is used as the final sSIMI score.

4.3 Datasets and Experimental Settings

The Zerospeech Challenge 2017 (Dunbar et al., 2017) provides five corpora to evaluate speech segmentation systems. These corpora are composed of speech recordings from different languages split into sentences using Voice Activity Detection. Three corpora (Mandarin, 2h30; English, 45h; French, 24h) are used for development, and two 'surprise' corpora for testing (German, 25h; Wolof, 10h). On each corpus, a separate SSE model from Algayres et al. (2022) is trained and a separate run of DP-Parse is performed to produce a full-coverage segmentation. DP-Parse's hyper-parameters from Table 2, as well as q's formula (Equation 5), were grid-searched to maximize token-F1 segmentation scores over the three development datasets. The two remaining test sets are used to show generalization of DP-Parse to new unseen languages. More details on hyper-parameter search is found in Appendix A.2.2.

For the ABX and sSIMI tasks, we train a BERT model as a Speech LM on the training set of the Librispeech dataset (Panayotov et al., 2015), composed of 960 hours of English recordings. We proceed by first training a SSE model from Algayres et al. (2022) on the Librispeech. Then, this latter dataset is segmented using DP-Parse. Each segmented speech sequence is aggregated into a single vector using the pre-trained SSE model. Segmented sentences are used to train a BERT model with masked language modeling and the Noise Contrastive Estimation (NCE) loss (Gutmann and Hyvärinen, 2010) (see Figure 4). The BERT model is composed of 12 layers, 12 attention heads per layer with 768 neurons each and a FFN size of 3072 neurons. To compute the NCE, two heads p and h are composed respectively of one fully connected linear layer with 256 neurons and two fully connected layers with ReLU activation with also 256 neurons. Batches are composed of utterances from a single speaker, and the 200 negative samples for the NCE are chosen within the batch. Fifteen percent of SSEs are masked in each batch during training. Only the BERT is trained, gradients are not propagated into the SSEs. As BERT is a multi-layer transformer model, the scores for ABX tasks and sSIMI are obtained by grid-searching the layer that performs best on the dev set of each task and evaluating that same layer on the corresponding test sets.

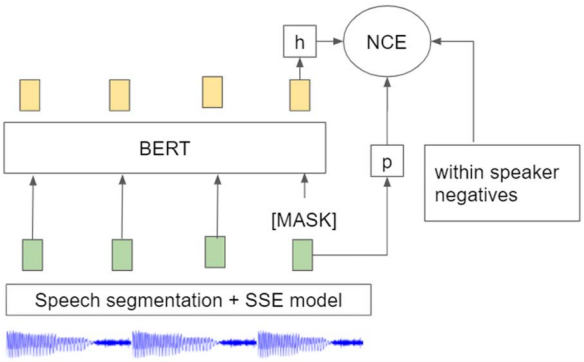

Figure 4: Our method to train a BERT model on SSEs. Speech is segmented into segments that are converted into vectors by a frozen SSE model. BERT is trained with the NCE loss. p and h are used to project vectors into a common space to compute the NCE objective. The negative samples are random SSEs from the same speaker.

	Mandarin (2h30)	French (24h)	English (45h)
DP Unigram	61m39s	240m44s	62m11s
DP-Parse	0m55s	25m12s	62m11s
Speed-up factor	67.3	9.6	9.8

Table 3: Time performances on 8 CPUs for the standard configuration of DP-Parse (10 iterations) and the Dirichlet Process Unigram (2000 iterations) implemented by Johnson et al. (2007).

The development and test sets for the ABX tasks are composed of triplets of sentences extracted from the training set of the Librispeech corpus. For sSIMI, the development and test set both contain two subsets: one with spoken words extracted from the Librispeech training set and one with spoken words synthesized with WaveNet (van den Oord et al., 2016).

5 Results

5.1 Results on Word-level Segmentation

Regarding text segmentation, DP-Parse compares favorably to the original Dirichlet Process Unigram model from Goldwater et al. (2009). It produces a 21-point increase in token F1 compared with the original version (and 3 points increase in boundary F score) for a 10× runtime speed-up (Table 3) under the Adaptor Grammar implementation of Johnson et al. (2007).

Regarding speech segmentation, we compare DP-Parse with the three best speech segmentation

Model	Token F1 ↑						Boundary F1 ↑					
	Dev			Test			Dev			Test		
	Mand.	French	Engl.	German	Wolof	*avg.*	Mand.	French	Engl.	German	Wolof	*avg.*
Speech data												
Every 120ms	10.7	11.7	9.8	5.4	12.4	10.0	52.4	49.4	48.2	41.3	55.0	49.2
Syllables$^{\times}$	4.4	5.1	4.1	3.0	4.2	4.2	49.1	46.3	43.7	38.1	40.6	43.6
SEA^{+}	12.1	8.3	8.6	7.5	14.8	10.3	52.2	48.4	46.1	40.9	54.2	48.4
ES-Kmeans$^{\vee}$	8.1	6.30	19.2	**14.5**	10.9	11.8	54.5	43.3	56.7	**52.3**	52.8	51.9
DP-Parse*	**16.0**	**15.3**	**21.9**	13.4	**17.5**	**16.8**	**59.3**	**55.9**	**60.0**	51.5	**59.0**	**57.1**
DP-Parse* + weak sup.	28.2	30.9	31.3	34.4	39.2	35	69.9	70.0	68.4	64.4	75.5	69.4
Text data												
DP Unigram‡	34.9	57.0	64.7	33.6	38.8	45.7	**79.5**	**86.1**	88.9	71.6	77.3	80.7
DP-Parse*	**50.0**	**68.1**	**78.5**	**67.4**	**69.1**	**66.6**	76.0	84.3	**89.8**	**83.5**	**84.1**	**83.5**

Table 4: Token and Boundary F1 scores (the higher, the better) for speech and text segmentation models across development and test set from the Zerospeech Challenge 2017. The average scores are computed over the five datasets. *: ours, $\times$: Räsänen et al. (2015), +: Bhati et al. (2020), $\vee$: Kamper et al. (2017b), ‡: Goldwater et al. (2009).

models submitted at the Zerospeech Challenge 2017 (Bhati et al., 2020; Kamper et al., 2017b; Räsänen et al., 2015). Table 4 reports the token F1 and boundary F1 obtained by these models over the 5 datasets of the Zerospeech Challenge 2017. Across all corpora, DP-Parse outperforms its competitors by at least 5 points in both boundary and token F1. We also introduce a naive baseline that draws boundaries every 120 milliseconds, disregarding the content of the speech signal. It turns out to be surprisingly competitive with all speech segmentation systems except DP-Parse. The latter is the only existing speech segmentation system that beats the naive baseline on all languages.

Most speech segmentation systems from the Zerospeech Challenge 2017 rely on off-the-shelf self-supervised representations of speech. The hope is that, without modification, these systems would benefit from future improvements in speech representations and mechanically lead to higher segmentation scores. Yet, such hopes have never been guaranteed by explicit experiments. Here, we test for the ability of DP-Parse to improve with better inputs. For that, we use the weakly supervised version of the SSE model from Algayres et al. (2022) trained with 10 hours of labelled speech data. On this type of input, DP-Parse doubles its token F1 score.

5.2 Results on Semantic and POS Metrics

In Table 5, the *segment-based* sections show how a BERT model can benefit from speech segmentation and SSE modeling on the tasks ABX_{sem}, ABX_{POS} and sSIMI. To do that, we trained BERT models along the pipeline depicted in Figure 4.

Let us first analyze the scores on the ABX tasks. The section *unsupervised, segment-based* shows that DP-Parse and two less-performant segmentation strategies lead to comparable ABX scores. Yet, the *weak supervision* section shows that by improving the quality of the SSEs with weak supervision, DP-Parse performs higher than other segmentation methods. The *partial supervision* section shows that with either perfect word boundaries or perfect segment representation (1-hot vectors), the ABX scores increase even more. Finally, the *full supervision* is regular text-based LM that serves as a topline model.

As baseline systems, we propose *frame-based* approaches that do not use speech segmentation nor SSEs. Speech-to-frames models like Wav2vec2.0, HuBERT, or CPC (Baevski et al., 2020; Hsu et al., 2021; van den Oord et al., 2019) are trained with masked language modeling in the spirit of text-based LM but on the raw speech signal. Therefore, these models should have already acquired some knowledge of semantics and POS-taggings. We evaluate speech-to-frames models directly on ABX_{sem}, ABX_{POS} and sSIMI. To do that, we simply skip the Speech LM as well as the segmentation and SSE steps from the method in Figure 3. Speech-to-frames models are used to encode the whole speech sentences from the ABX triplets, and the frames within the word boundaries provided by the ABX tasks are mean-pooled to form the HLSEs. The results show that Wav2vec2.0 Large and HuBERT Large are

Input Representations	Segments	SSE	BERT	Dev Set $ABX_{sem}\uparrow$	Dev Set $ABX_{POS}\uparrow$	Test Set $ABX_{sem}\uparrow$	Test Set $ABX_{POS}\uparrow$	Dev Set sSIMI ↑	Test Set sSIMI ↑
No supervision, frame based									
CPC^{+}	–	–	–	51.22 [2]	53.67 [2]	53.86 [2]	54.68 [2]	**6.14 [2]**	4.34 [2]
W2V2 base$^{\times}$	–	–	–	54.96 [8]	55.78 [8]	56.20 [8]	56.86 [8]	3.85 [9]	5.21 [9]
HuBERT base‡	–	–	–	54.65 [8]	55.65 [8]	55.12 [8]	56.77 [8]	3.53 [8]	0.68 [8]
W2V2 large$^{\times}$	–	–	–	57.69 [13]	**59.86 [13]**	58.88 [13]	**60.93 [13]**	2.04 [11]	**6.82 [11]**
HuBERT large‡	–	–	–	**58.04 [23]**	57.77 [23]	**58.95 [23]**	57.19 [23]	4.14 [7]	0.55 [7]
No supervision, segment based									
W2V2 base [8]$^{\times}$	120ms	unsup†	✓	59.65 [7]	**70.66 [7]**	58.64 [7]	**71.68 [7]**	4.10 [9]	**4.19 [9]**
W2V2 base [8]$^{\times}$	Syllables$^{\vee}$	unsup†	✓	59.09 [6]	68.58 [6]	58.49 [6]	68.81 [6]	0.96 [9]	3.48 [9]
W2V2 base [8]$^{\times}$	DP-Parse	unsup†	✓	**61.09 [8]**	68.15 [7]	**62.73 [8]**	69.35 [7]	**4.54 [5]**	3.49 [5]
Weak supervision, segment based									
W2V2 base [8]$^{\times}$	120ms	weak-sup†	✓	59.64 [8]	70.04 [8]	59.43 [8]	70.58 [8]	**5.83** [7]	**5.81 [7]**
W2V2 base [8]$^{\times}$	Syllables$^{\vee}$	weak-sup†	✓	61.44 [6]	70.89 [7]	60.28 [6]	71.03 [7]	0.51 [9]	0.79 [9]
W2V2 base [8]$^{\times}$	DP-Parse	weak-sup†	✓	**64.24 [8]**	**72.65 [7]**	**64.82 [8]**	**72.67 [7]**	5.19 [9]	4.51 [9]
Partial supervision, segment based									
W2V2 base [8]$^{\times}$	gold words	unsup†	✓	67.89 [5]	75.46 [7]	67.17 [5]	75.61 [7]	6.82 [5]	6.30 [5]
W2V2 base [8]$^{\times}$	gold words	weak-sup†	✓	71.38 [5]	77.75 [6]	71.04 [5]	77.64 [6]	2.72 [5]	6.83 [5]
text	DP-Parse	1-hot	✓	**75.38 [4]**	**78.57 [4]**	**74.89 [4]**	**78.49 [4]**	**21.42 [2]**	**19.54 [2]**
Full supervision, segment based									
text	gold words	1-hot	✓	**83.23 [3]**	**81.07 [3]**	**84.12 [3]**	**80.09 [3]**	**41.78 [1]**	**36.83 [1]**

Table 5: Semantic and Part-of-Speech discrimination scores (the higher, the better). Segment-based models compute SSEs for each segment (obtained by speech segmentation) on which a BERT model is trained with a NCE loss. Frame-based models are pre-trained baseline models without speech segmentation nor BERT training. sSIMI scores are average over the Librispeech and synthetic subsets. Layers used are given between brackets. $+$:van den Oord et al. (2019), $\times$:Baevski et al. (2020), ‡:Hsu et al. (2021), $\vee$: Räsänen et al. (2015), †: Algayres et al. (2022).

strong baseline systems for our ABX metrics. Yet, they score in average below the *segment-based* approaches.

Regarding the scores obtained on sSIMI from Nguyen et al. (2020), the table of results (Table 5) shows important downsides on this task compared to our ABX tasks. First, the scores sometimes show large inconsistencies across development and test sets, which is not the case for the ABX tasks. Second, while our ABX tasks are sensitive to improvements in speech segmentation and SSE modeling, sSIMI is not and show comparable scores across most sections of Table 5. One reason for that could be that self-supervised speech systems (Wav2vec2.0, HuBERT, and CPC) as well as SSE models from Algayres et al. (2022) are not trained to encode short sequence of speech, especially extracted as chunks from a sentence.

6 Conclusion and Open Questions

We introduce a new speech segmentation pipeline that sets the state-of-the-art on the Zerospeech's datasets at 16.8 token F1. The whole pipeline needs no specific tuning of hyper-parameters, making it ready to use on any new languages. We showed that the problem of speech segmentation can be reduced to the problem of learning discriminative speech representations. Indeed, using different level of supervision, our pipeline reaches up to 35 token F1 score. Therefore, as long as the field of unsupervised representation learning makes headway, this method should automatically produce higher token F1 scores.

A first avenue of improvement of the current work is in the SSE component. Here, we took the system described in Algayres et al. (2022) out of the box, and we showed good performance in speech segmentation compared to the state of the art, but there was still a large margin of improvement compared to text-based system. A recent unpublished paper (Kamper, 2022) came to our attention based on the non-lexical principle and showed similar or slightly better results than ours on a subset of the ZR17 language. Kamper (2022) also uses a segmentation lattice that resembles ours for inference. Yet, as we have shown, our system is monotonous with input embedding quality, and can therefore reach much better performance if the speech sequence embedding component were better trained. Further work is needed to improve the SSE component based on purely unsupervised methods.

Despite our computational speedup, our current system would face challenges to scale up the approach to larger datasets than Librispeech. While it uses FAISS, an optimized search library, it is unclear that storing each instance of a possible

segment is necessary. One interesting avenue of research would therefore be to downsample both instance lexicons to alleviate space complexity and/or to trade storage with compute by recalculating the embeddings on-line.

Another interesting improvement would be to compute bigram probabilities instead of unigram probabilities, as previously explored in Goldwater et al. (2009). Yet, this possibility looks challenging, as explained in Appendix A.2.3.

Finally, our method showed promising results in terms of semantic and POS encoding when taken as a preprocessor for a language model. Such a phenomenon is displayed by our new in-context semantic tasks, but not by a previous without-context semantic task from Nguyen et al. (2020). Further work is needed to show whether this can translate into high quality generations when used as a component to generative systems (Lakhotia et al., 2021).

A Appendix

A.1 Construction of Triplets for ABX_{sem} and ABX_{POS}

Let's write the series of triplet from our ABX tasks: $(A_i, B_i, X_i)_{0 \leq i < N}$. For all i, A_i is defined as a tuple $(R_i^a, s_i^a, e_i^a, t_i^a)$ where R_i^a is a recording of a whole sentence and s_i^a and e_i^a are the temporal boundaries of a word with phonetic transcription t_i^a. B_i and X_i are defined identically. The sentences are extracted from the Librispeech dataset, a 960-hour corpus of read English literature.

The Speech LM is asked to encode the whole sentences (R_i^a, R_i^b, R_i^x) and compute the three word embeddings for the words of interest using the provided timestamps. The ABX score is given by the following formula:

$$ABX(T, d) = \frac{1}{N} \sum_{i=0}^{N-1} \quad d(f_i^a, f_i^x) < d(f_i^b, f_i^x)$$

where T is a collection of triplet word embeddings $(f_i^a, f_i^b, f_i^x)_{0 \leq i < N}$ and d is the cosine distance.

ABX_{sem} is composed of 502 pairs (evenly split into a development and test set) of words created using a synonym dictionary. By sampling a list distractor for each pair, we reached 1557 different triplets. For each unique word found in the triplets, we sample 10 occurrences from the sentences of the Librispeech corpus. In total, ABX_{sem} is computed over 1.5M triplets. ABX_{POS} is composed of 9997 pairs of words built from the *WordNet* database (Fellbaum, 1998). The words in each pair have the same POS tags that can be either noun, verb, or adjective. By sampling distractors and 10 occurrences of each word types, we reached 37.3M triplets.

	Token-F1	
Languages	DP-Parse + k-means	DP-Parse + k-NN
mandarin	0.1	0.147
french	0.07	0.153
english	0.08	0.219

Table 6: Token-F1 scores obtained by DP-Parse on three corpora from the Zerospeech 2017. The frequency estimation is performed by either a k-NN or by a k-means that knows the true number of clusters (found in the transcription of those datasets).

A.2 Ablation Study

A.2.1 k-means Instead of k-NN

To estimate DP-Parse parameters (L_{0_w} and L_w), we followed a non-clustering approach with a k-NN density estimation. In this section, instead, we use k-means clustering to estimate DP-Parse parameters. We used the transcriptions of the development sets from the Zerospeech Challenge 2017 to give to the k-means the true number of clusters is it suppose to find when clustering SSEs from L_0 and L. The value of L_{0_w} and L_w are given by the size of the cluster in which w is found. From Table 6, the segmentation scores obtained using k-means are much lower than those obtained by k-NN. Indeed, as shown in a study by Algayres et al. (2020), k-means is subject to the *uniform effect* (Wu, 2012), which makes it not suited to estimate frequencies on highly skewed distributions, as the distribution of word types which follows the Zipf Law (Zipf, 1949).

A.2.2 DP-Parse Hyper-parameters

In Table 7, we provide DP-Parse speech segmentation performance, as measured by token-F1 scores, over three datasets (Mandarin, English, and French) for different hyper-parameters values. Surprisingly, increasing the number of neighbors (k), the number of samples in L_0 ($\#L_0$), or the beam size does not improve token-F1 scores. DP-Parse time complexity scales linearly with each of these three parameters, therefore we keep their values as low as possible.

	Token-F1			
	Mandarin	French	English	*average*
$k = 10$	16.8	14.7	21.5	*17.7*
$\boldsymbol{k = 100}$	**16.7**	**15.3**	**21.9**	***18.0***
$k = 200$	16.9	15.4	21.9	*18.0*
$beam = 5$	16.6	15.4	21.9	*18.0*
$\boldsymbol{beam = 10}$	**16.7**	**15.3**	**21.9**	***18.0***
$beam = 100$	16.8	15.2	21.9	*18.0*
$\#L_0 = 300k$	16.9	14.7	21.5	*17.7*
$\boldsymbol{\#L_0 = 1M}$	**16.7**	**15.3**	**21.9**	***18.0***
$\#L_0 = 3M$	16.6	15.4	22.0	*18.0*
$\alpha_0 = 10^{-4}$	16.4	15.0	21.4	*17.6*
$\alpha_0 = 10$	16.3	15.3	21.7	*17.8*
$\boldsymbol{\alpha_0 = 10^2}$	**16.7**	**15.3**	**21.9**	***18.0***
$\alpha_0 = 10^3$	17.0	15.1	21.7	*17.9*
$\alpha_0 = 10^5$	16.3	15.0	19.7	*17.0*
$\delta = 1$	9.0	13.4	14.5	*12.3*
$\delta = 3$	17.0	14.8	21.7	*17.8*
$\boldsymbol{\delta = 4}$	**16.7**	**15.3**	**21.9**	***18.0***
$\delta = 5$	15.8	15.0	21.4	*17.4*
$\delta = 9$	15.2	14.8	19.9	*16.6*
$\gamma = 0$	14.7	14.7	19.9	*16.4*
$\gamma = 1.6$	16.0	15.0	21.7	*17.6*
$\boldsymbol{\gamma = 1.8}$	**16.7**	**15.3**	**21.9**	***18.0***
$\gamma = 2$	16.9	14.9	21.6	*17.8*

Table 7: Token F1-score on the number of neighbors for the k-NN search (k), the concentration parameter ($alpha_0$), the beam-search size ($beam$), the number of samples in L_0 ($\#L_0$), and the pair (δ, γ) from the penalty function. The default parameter values are $k = 100$, $\alpha_0 = 100$, $beam = 10$, $\#L_0 = 1M$, $\gamma = 1.8$, and $\delta = 4$ (also in bold in the table).

Another unexpected result is the low impact of the concentration parameter (α_0). The value of this parameter should be a crucial as it controls the implicit vocabulary by controlling the amount of rare words in the segmentation. This observation has led us to create the penalty function q from Equation 5 that offers a control on word-lengths, which also impacts the implicit vocabulary. This time, we noticed that q's shape strongly impact token-F1 scores. By a careful tuning of the penalty function, we increased the token-F1 from 16.4, with $\gamma = 0$ (i.e., no penalty function) to 18, with $\gamma = 1.8$ and $\delta = 4$.

A.2.3 Hierarchical Dirichlet Process

Goldwater's bigram Hierarchical Dirichlet Process (bigram-HDP), presented in Goldwater et al. (2009) for text segmentation computes the probability of two consecutive word-candidate, $P(\langle w_{i-1}, w_i \rangle)$, to be segmented instead of only one, $P(w_i)$, as in the unigram Dirichlet process (unigram-DP). In Goldwater et al. (2009), the computation of the bigram-HDP probability $P(\langle w_{i-1}, w_i \rangle)$ requires the number of different *types* that can be found in all previously segmented bigrams with w_i as second word. This is particularly difficult to adapt to speech segmentation with DP-Parse, because counting types require an explicit clustering step. We doubt that using k-means would work as we have shown that clustering SSEs with k-means works poorly, even if the true number of clusters is known (see Table 6). More advanced clustering technics could work better than k-means as shown in Kamper et al. (2014)—for example, Chinese Whispers Clustering, Hierarchical K-means, or probabilistic clustering using GMMs, yet it would require a large effort to incorporate it to DP-Parse.

References

Robin Algayres, Adel Nabli, Benoit Sagot, and Emmanuel Dupoux. 2022. Speech sequence embeddings using nearest neighbors contrastive learning.

Robin Algayres, Mohamed Salah Zaiem, Benoît Sagot, and Emmanuel Dupoux. 2020. Evaluating the reliability of acoustic speech embeddings. https://doi.org/10.21437/Interspeech.2020-2362

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California. Association for Computational Linguistics.

Saurabhchand Bhati, Jesús Villalba, Piotr Żelasko, and Najim Dehak. 2020. Self-expressing autoencoders for unsupervised spoken term discovery.

Saurabhchand Bhati, Jesús Villalba, Piotr Żelasko, Laureano Moro-Velazquez, and Najim Dehak.

2021. Segmental contrastive predictive coding for unsupervised word segmentation.

Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. 2019. pyannote.audio: neural building blocks for speaker diarization. https://doi.org/10.1109/ICASSP40776.2020.9052974

Michael Brent and Timothy Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125. https://doi.org/10.21437/Interspeech.2011-304

Michael Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky. 2011. Rapid evaluation of speech representations for spoken term discovery. In *INTERSPEECH*, pages 821–824. https://doi.org/10.21437/Interspeech.2011-304

Yu-An Chung and James Glass. 2018. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. https://doi.org/10.21437/Interspeech.2018-2341

Santiago Cuervo, Maciej Grabias, Jan Chorowski, Grzegorz Ciesielski, Adrian Łańcucki, Paweł Rychlikowksi, and Ricard Marxer. 2021. Contrastive prediction strategies for unsupervised segmentation and categorization of phonemes and words. https://doi.org/10.1109/ICASSP43922.2022.9746102

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding.

Ewan Dunbar, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, Laurent Besacier, Xavier Anguera, and Emmanuel Dupoux. 2017. The zero resource speech challenge 2017. https://doi.org/10.1109/ASRU.2017.8268953

Ewan Dunbar, Julien Karadayi, Mathieu Bernard, Xuan-Nga Cao, Robin Algayres, Lucas Ondel, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux. 2020. The zero resource speech challenge 2020: Discovering discrete subword and word units. https://doi.org/10.21437/Interspeech.2020-2743

R. Eskander, Owen Rambow, and Tianchun Yang. 2016. Extending the use of adaptor grammars for unsupervised morphological segmentation of unseen languages. In *COLING*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

Pierre Godard, Laurent Besacier, François Yvon, Martine Adda-Decker, Gilles Adda, Hélène Maynard, and Annie Rialland. 2018. Adaptor grammars for the linguist: Word segmentation experiments for very low-resource languages. In *Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 32–42, Bruxelles, Belgium. https://doi.org/10.18653/v1/W18-5804

S. Goldwater, Tom Griffiths, and M. Johnson. 2009. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112:21–54. https://doi.org/10.1016/j.cognition.2009.03.008

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney, Australia. Association for Computational Linguistics. https://doi.org/10.3115/1220175.1220260

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, PP:1–1.

Yushi Hu, Shane Settle, and Karen Livescu. 2020a. Acoustic span embeddings for multilingual query-by-example search.

Yushi Hu, Shane Settle, and Karen Livescu. 2020b. Multilingual jointly trained acoustic and written word embeddings. pages 1052–1056.

Christiaan Jacobs, Yevgen Matusevych, and Herman Kamper. 2021. Acoustic word embeddings for zero-resource languages using self-supervised contrastive learning and multilingual adaptation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 919–926. https://doi.org/10.1109/SLT48900.2021.9383594

Aren Jansen and Benjamin Van Durme. 2011. Efficient spoken term discovery using randomized algorithms. In *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, pages 401–406. https://doi.org/10.1109/ASRU.2011.6163965

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 641–648. MIT Press.

Herman Kamper. 2018. Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models. *CoRR*, abs/1811.00403.

Herman Kamper. 2022. Word segmentation on discovered phone units with dynamic programming and self-supervised scoring.

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2017a. A segmental framework for fully-unsupervised large-vocabulary speech recognition. *Computer Speech & Language*, 46:154–174. https://doi.org/10.1016/j.csl.2017.04.008

Herman Kamper, Aren Jansen, Simon King, and Sharon Goldwater. 2014. Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 100–105. https://doi.org/10.1109/SLT.2014.7078557

Herman Kamper, Karen Livescu, and Sharon Goldwater. 2017b. An embedded segmental k-means model for unsupervised segmentation and clustering of speech. *CoRR*, abs/1703.08135. https://doi.org/10.1109/ASRU.2017.8269008

Herman Kamper and Benjamin van Niekerk. 2020. Towards unsupervised phone and word segmentation using self-supervised vector-quantized neural networks. https://doi.org/10.21437/Interspeech.2021-50

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2019. Learning to discover, ground and use words with segmental neural language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6441, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1645

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Adelrahman Mohamed, and Emmanuel Dupoux. 2021. Generative spoken language modeling from raw audio.

Chia-ying Lee and James Glass. 2012. A nonparametric Bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40–49, Jeju Island, Korea. Association for Computational Linguistics.

Chia-ying Lee, Timothy J. O'Donnell, and James Glass. 2015. Unsupervised lexicon discovery from acoustic input. *Transactions of the Association for Computational Linguistics*, 3:389–403. https://doi.org/10.1162/tacl_a_00146

Haizhou Li and Baosheng Yuan. 1998. Chinese word segmentation. In *Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation*, pages 212–217, Singapore. Chinese and Oriental Languages Information Processing Society.

Bogdan Ludusan, Maarten Versteegh, Aren Jansen, Guillaume Gravier, Xuan-Nga Cao, Mark Johnson, and Emmanuel Dupoux. 2014. Bridging the gap between speech technology and natural language processing: An evaluation toolbox for term discovery systems. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 560–567, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

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. https://doi.org/10.3115/1687878.1687894

Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Evgeny Kharitonov, Alexei Baevski, Ewan Dunbar, and Emmanuel Dupoux. 2020. The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling. *CoRR*, abs/2011.11588.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*. https://doi.org/10.1109/ICASSP.2015.7178964

Alex S. Park and James R. Glass. 2007. Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):186–197. https://doi.org/10.1109/TASL.2007.909282

Emanuel Parzen. 1962. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3). https://doi.org/10.1214/aoms/1177704472

Rachid Riad, Corentin Dancette, Julien Karadayi, Neil Zeghidour, Thomas Schatz, and Emmanuel Dupoux. 2018. Sampling strategies in siamese networks for unsupervised speech representation learning. https://doi.org/10.21437/Interspeech.2018-2384

Okko Räsänen, Gabriel Doyle, and Michael Frank. 2015. Unsupervised word discovery from speech using automatic segmentation into syllable-like units. https://doi.org/10.21437/Interspeech.2015-645

Thomas Schatz, Vijayaditya Peddinti, Francis Bach, Aren Jansen, Hynek Hermansky, and Emmanuel Dupoux. 2013. Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline. In *INTERSPEECH 2013 : 14th Annual Conference of the International Speech Communication Association*, pages 1–5, Lyon, France. https://doi.org/10.21437/Interspeech.2013-441

Shane Settle and Karen Livescu. 2016. Discriminative acoustic word embeddings: Recurrent neural network-based approaches. https://doi.org/10.1109/SLT.2016.7846310

Alexis Thual, Corentin Dancette, Julien Karadayi, Juan Benjumea, and Emmanuel Dupoux. 2018. A K-nearest neighbours approach to unsupervised spoken term discovery. In *IEEE Spoken Language Technology SLT-2018*, Proceedings of SLT 2018. Athènes, Greece. https://doi.org/10.1109/SLT.2018.8639515

Maarten Versteegh, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux. 2016. The zero resource speech challenge 2015: Proposed approaches and results. In *SLTU-2016 Procedia Computer Science*, volume 81, pages 67–72. ISCA-ITRW. https://doi.org/10.1016/j.procs.2016.04.031

Junjie Wu. 2012. *The Uniform Effect of K-means Clustering*.

George Kingsley Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press.