DEEP LEARNING

Text Segmentation - Approaches, Datasets, and Evaluation Metrics

In this post, we give an overview of the best approaches, datasets, and evaluation metrics commonly used for the task of Text Segmentation.

Taufiquzzaman Peyash
Deep Learning Engineer

Nov 16, 2021

Segmentation

Text Segmentation is the task of splitting text into meaningful segments. These segments can be composed of words, sentences, or topics. In this post, we look at a specific type of Text Segmentation task - Topic Segmentation, which divides a long body of text into segments that correspond to a distinct topic or subtopic.

For example, consider an hour-long podcast transcription generated by an Automatic Speech Recognition (ASR) system. The transcription can be lengthy and, as a result it can be easy to lose track of which sentence you're currently reading. Automatic Topic Segmentation solves this problem by dividing the text into multiple segments, making the transcription more readable.

Figure: Topic Segmentation

Types of Text Formats

There are different types of text we may want to segment. For example:

- Written text like blogs, articles, news, etc.
- Transcription of TV news, where a single person is talking
- Transcription of a podcast where more than one person is talking
- Transcription of phone calls (e.g., call center)
- Transcription of an online meeting where many people are talking

The above list is ordered based on the level of noise the text may contain (i.e., typos, grammatical errors, or incorrect ...e of words in the case of automatic transcription). Noise is an

contributes to the quality of the segments predicted by the Topic Segmentation models. We will discuss this more in later sections.

Since blogs and articles are mostly typed on a computer, they contain the least amount of noise.

Transcriptions of online meetings, for example, contain the highest level of noise since there may be several people speaking with different quality microphones, with different accents and over varying internet connection quality, which can cause accuracy issues with ASR systems. On the other hand, TV News Reports and Podcasts are usually recorded with studio-quality microphones, so there is far less noise compared to online meetings, which results in much more accurate text transcriptions.

Experience

Processing with AssemblyAI

Try our Speech-to-Text API with advanced features like topic detection and auto chapters - perfect for implementing text segmentation in your projects.

Try AssemblyAI Playground

Use Cases for Topic Segmentation

Now that we know what Topic Segmentation is, and the types of text formats we are working with, here are some real-life use cases:

- **Readability**: One of the main reasons we want to split text into multiple paragraph-sized segments is readability. Consider

long string of text makes it much harder to read. An Automatic Topic Segmentation model can be used to break down long, continuous text into bite-sized information so it is easier for the reader to consume.

- **Summarization**: Text Summarization helps the reader to summarize any given text. However, if the text deals with multiple topics, it can be challenging for the Text Summarization model to capture all the topics in the summary. A Topic Segmentation model can make it easier to generate a summary for each segment in bullet points, covering all the topics or themes in a given text.

- **Turning News Reports into Articles:** Today, news reports are often distributed through multiple channels that require different

written formats (like articles or blog posts). To reuse a news report video as an article or blog, we can transcribe the video using a Speech-to-Text API like the one we build at AssemblyAI, and apply Topic Segmentation to the transcription. This will help to organize the article into a more readable format for the readers.

- **Information Retrieval:** Given a large amount of text documents, we can cluster text that belongs to the same topic. Once our documents are segmented by topic, we can easily extract the information we need from each document.
- **AI Writing Assistant:** For AI writing assistants like Grammarly, Topic Segmentation can be used to suggest to writers when to start a new paragraph, making their writing more readable.

Implement Text Segmentation?

Sign up for AssemblyAI and start building powerful NLP applications with our state-of-the-art API.

Start Building with AssemblyAI

Methods for Topic Segmentation and Evaluation Metrics

Before jumping into discussing the existing methods for Automatic Topic Segmentation, let's first investigate a potential solution to the topic segmentation problem and look at evaluation metrics for segmentation accuracy.

Recap, in Automatic Topic Segmentation, our goal is to segment

would classify each sentence in a document and determine whether it is a boundary sentence (i.e., the last sentence of a paragraph). In other words, we can think of Topic Segmentation as a binary classification problem, where we classify each sentence and determine if it is a boundary sentence.

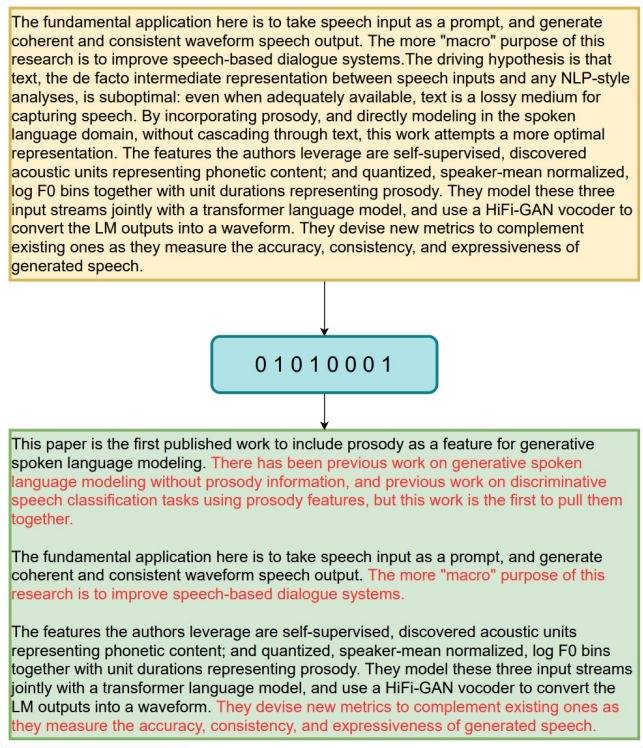

Figure: Topic Segmentation as Binary Classification

Segmentation problem, let's discuss its evaluation metrics. The most commonly used metrics are:

- Precision & Recall
- Pk
- WindowDiff

Precision & Recall

Since this is a binary classification problem, you might be tempted to use Precision & Recall as an evaluation metric. However, there are some challenges with using Precision & Recall for this type of classification problem. Let's first understand what Precision & Recall mean in relation to Topic Segmentation.

Precision: percentage of boundaries identified by the model that are true boundaries

Recall: percentage of true boundaries

Precision & Recall is that they are not sensitive to near misses. To understand what a near miss is, let's consider two Topic Segmentation models A-0 and A-1. In the following figure, the **Ref** is the ground truth and each of the blocks represent a sentence. The vertical lines indicate a topic/subtopic segmentation boundary.

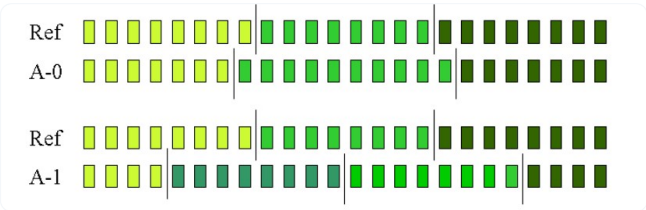

Figure: Ground truth and output of segmentation algorithms

From the figure, you can clearly see the prediction made by model A-0 is pretty close to the ground truth. This is called a near miss, where the prediction is off by one or two sentences. On the other hand, the prediction made by model A-1 is pretty

model A-0 should be penalized significantly less than model A-1. But that is not the case, both models get the same Precision & Recall score. This is because Precision & Recall do not consider how close or far away the boundary predictions are.

Pk Score

To solve the challenges with Precision & Recall, the Pk score was introduced by Beeferemen et al (1997).

Pk is calculated by using a sliding window-based method. The window size is usually set to half of the average true segment number. While sliding the window, the algorithm determines whether the two ends of the window are in the same or different segments in the ground truth segmentation, and increases a counter if there is a mismatch. The final score is calculated by scaling the penalty between 0 and 1 and dividing the number of

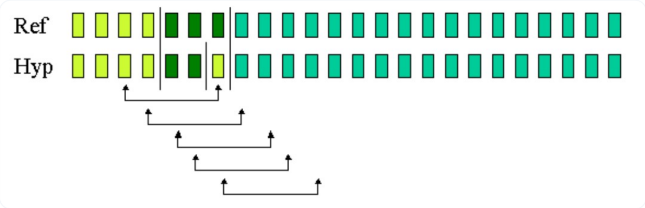

Figure: Sliding window over reference and predictions (Pevzner and Hearst - 2002)

A model that predicts all the boundaries correctly gets a score of 0. So the lower the score, the better.

Challenges with the Pk Evaluation Metric:

- False negatives are penalized more than false positives.
- Does not take the number of boundaries into consideration. If there are multiple boundaries inside the window, Pk doesn't consider that.
- Sensitive to the variation in segment size.

WindowDiff

WindowDiff was introduced to solve the challenges with the Pk score. This is also calculated by a sliding window. In this case, for each position of the window of size *k*, we simply compare how many boundaries are in the ground truth, and how many boundaries are predicted by the Topic Segmentation model.

$$WindowDiff(ref, hyp) = \frac{1}{N-k}\sum_{i=1}^{N-k}(|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0)$$

Figure: Equation for calculating WindowDiff score

Here, `b(i, j)` is a function that returns the number of boundaries between two positions `i` and `j` in the text. *N* represents the number of sentences in the text.

In practice, both Pk and WindowDiff

closer to the actual boundaries. For a more detailed comparison of these metrics and exactly how WindowDiff score solves the challenges with Pk, you can refer to Pevzner et al (2002).

Methods for Topic Segmentation

In this section, we take a look at the most common methods of Topic Segmentation, which can be divided into mainly two groups - Supervised & Unsupervised.

Supervised Approaches

Supervised approaches are pretty straightforward - We take a labelled dataset and then we try to fit a model on it. This works well when we have a domain-specific segmentation task and the dataset belongs to the same domain. For example, if we know that model will be seeing texts similar

produce the best result. However, it will perform worse if used in other domains, for example on a news article or transcriptions of meetings.

In the supervised approach, we want to classify each sentence to determine whether it is a boundary sentence or not. Here is how the pipeline works at high level:

1. Take text as input
2. Extract all the sentences from the text, i.e., segment the text into sentences. (You can use libraries like `nltk`, `stanza`, `trankit` etc. for this task)
3. Classify each sentence--this will be a binary classification

One important thing to note is that we're taking input at a word level and predicting on a sentence level. This means we need to convert the word-level representation (embeddings) to

sentence to get an aggregated representation. This will give us sentence-level embeddings.

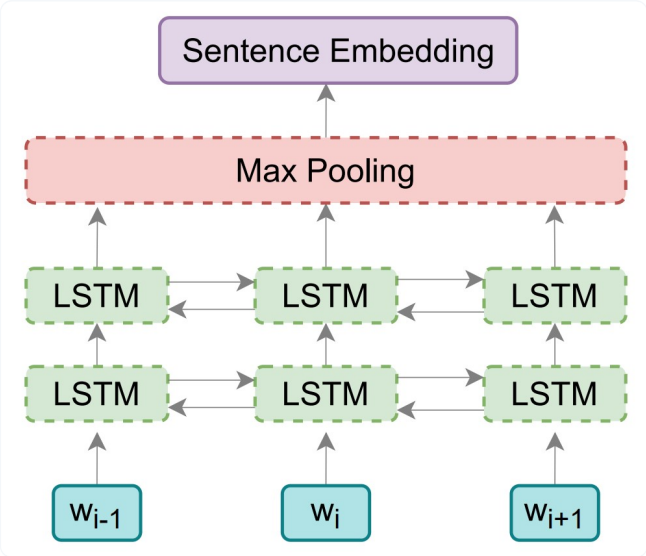

Figure: Calculation of sentence embeddings from word embedding

Additionally, if we use a model like BERT for this, we can get the embedding of the `[CLS]` token instead of aggregating all the word embeddings in a sentence. This is because in BERT, the `[CLS]` token aggregates the representation of the ...le sequence.

and calculate softmax over the output. Using a bidirectional LSTM/ Transformer is a good idea here because it will enable the model to look at both the left and right context of the sentence before making a decision.

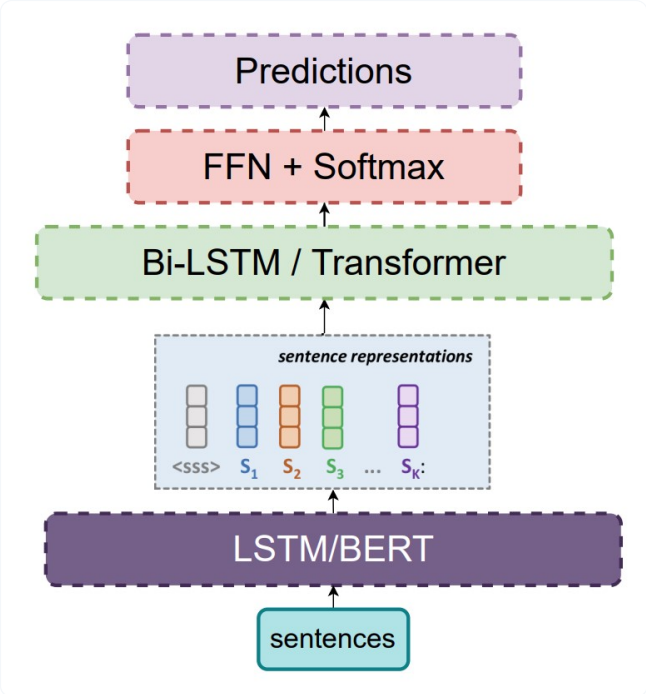

Figure: Boundary sentence classifier model

The LSTM based approach described above is actually used in Koshorek et al., which achieved a 22.13 Pk score on

Transformer-based model which holds the current State-of-the-Art result on datasets like Wiki-727k, CHOI, Elements & Cities etc. They used two-level Transformers: one at token-level and another one at sentence-level. On top of it, the prediction objective was augmented with an auxiliary coherence modeling objective. The key idea is that text coherence is related to text segmentation. This means a text within a segment is expected to be more coherent than the text in a different segment.

Unsupervised Approaches

Unsupervised approaches neither have a learning phase nor labelled data. Therefore, unsupervised approaches leverage different techniques for Topic Segmentation, such as:

- Lexical Cohesion

- Graph
- Similarity Measurement

We will cover how these approaches work at a high level.

Lexical Cohesion

A group of words is “lexically cohesive” if they are semantically related. The level of lexical cohesion is determined by the lexical frequency and distribution, and there are algorithms that exploit lexical cohesion to segment the text. The idea is that when there is a subtopic shift, the lexical cohesion will be lower between two blocks of text, and we can segment the text based on that.

TextTiling: TextTiling was introduced by Hearst (1997) and is one of the first unsupervised topic segmentation algorithms. It's a moving window-ed approach that uses lexical cohesion between blocks of text to

The algorithm has three main components:

- First, it divides the input text into sequences of relevant tokens and calculates the cohesion at each potential boundary point.
- It then uses these cohesion scores to produce depth scores for each potential boundary point that has a lower cohesion than the neighboring boundary points.
- Using these depth scores, the algorithm is able to select boundary points where the depth is low relative to the other depth scores, indicating that the gap represents a topic shift in the text.

LCseg: LCseg was introduced by Galley et al. (2003) for segmenting multiparty conversations. The algorithm uses lexical cohesion to ment topics, and it can handle both

- A method to identify and weight strong term repetitions using lexical chains.
- A method to hypothesize topic boundaries given the knowledge of multiple, simultaneous chains of term repetitions.

Topic Modeling

TopicTiling: This algorithm is similar to TextTiling except it uses a Latent Dirichlet Allocation (LDA) topic model for segmentation. Conceptually, it's simpler since it does not have to account for the sparsity of word-based features. Topic IDs generated by an LDA Topic Model are used instead of words.

Graph Based

GraphSeg: Glavas et al. (2016)

oses a graph-based algorithm to directly capture the semantic

instead of approximating it with topical similarity. Each node of the graph is a sentence and edges are created for pairs of semantically related sentences. Coherent segments are then determined by finding maximal cliques of the relatedness graph.

Similarity Based

Cosine Similarity: A window-based approach similar to TextTiling can be used to measure cosine similarity between two blocks of text. And the text can be represented with BERT contextual embeddings, which works significantly better than a simple bag-of-words or word2vec embeddings. For example, Solbiati et al (2021) uses embeddings from sentence-BERT, which has a siamese and triplet network structure and provides richer sentence-level embeddings.

Datasets

TDT Corpus (1998)

- Topic Detection and Tracking (TDT) is a DARPA-sponsored initiative to investigate the State-of-the-Art results in Topic Detection.
- The dataset has 16,000 news stories - half taken from Reuters newswire, and the other half taken from a CNN broadcast news transcript.
- Dataset: https://catalog.ldc.upenn.edu/LDC98T25

CHOI (2000)

- This is a synthetic dataset generated from the Brown corpus, basically by concatenating random documents.
- Contains 700 documents.

- Dataset: https://github.com/koomri/text-segmentation/tree/master/data/choi

Galley Dataset (2003)

- Also an artificially generated corpus.
- Contains two corpora, each one having 500 documents.
- Compared to CHOI, the segments vary from 4 to 22 segments.

Elements and Cities (2009)

- Chen et al. (2009) created these two small datasets from Wikipedia-based cities and elements.
- Commonly used for the evaluation of Topic Segmentation models.

i727k (2018)

- Wiki727k contains 727,746 English documents covering a wide variety of topics.
- Dataset: https://github.com/koomri/text-segmentation

Malach Corpus (2019)

- This corpus consists of over 115,000 hours of natural speech from 52,000 speakers in 32 different languages.
- 10% of this dataset has been manually segmented for the Topic Segmentation task.
- Dataset: https://catalog.ldc.upenn.edu/LDC2019S11

QMSum (2021)

- This dataset is mainly for text summarization tasks, constructed from a transcribed version of the

segmented based on topic or subtopic shift.

- This is a very good dataset if you want to evaluate your Topic Segmentation model against spoken text like dialogue, conversation, and meetings.
- Dataset: https://github.com/Yale-LILY/QMSum

Key Takeaways

- Segmenting text based on topics or subtopics can significantly improve the readability of text, and makes downstream tasks like summarization or information retrieval much easier.
- The main ways to evaluate a Text Segmentation model is through the Precision & Recall, Pk, and WindowDiff evaluation metrics.
- Depending on the task at hand,

methods provide viable options for building good performing Text Segmentation models.

References

- Discourse Coherence in the Wild: A Dataset, Evaluation and Methods
- Advances in domain-independent linear text segmentation
- Discourse Segmentation of Multi-Party Conversation

Structure using Latent Permutations

- Text Segmentation as a Supervised Learning Task
- Automated Transcription and Topic Segmentation of Large Spoken Archives
- QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization
- Text Segmentation as a Supervised Learning Task
- Statistical Models for Text Segmentation
- A Critique and Improvement of an Evaluation Metric for Text Segmentation
- Two-Level Transformer and Auxiliary Coherence Modeling for Improved Text Segmentation
- TextTiling: Segmenting Text into Multi-paragraph Subtopic

Party Conversation

- Unsupervised Text Segmentation Using Semantic Relatedness Graphs
- Unsupervised Topic Segmentation of Meetings with BERT Embeddings

Popular posts

INDUSTRY Jan 9, 2025

Top 6 benefits of integrating LLMs for Conversat Intelligenc platforms

Kelsey Foster Growth

ANNOUNCEMENTS Ja 9, 20

Dev.to x AssemblyA Winter Speech-to-Text Challenge Winners

Kelsey Foster Growth

INDUSTRY Jan 9, 2025

Top AI models for conversati intelligenc

Kelsey Foster Growth

INDUSTRY Dec 19, 2024

What is voice intelligenc and how does it work?

Jesse Sumrak Featured writer

Products

Core Transcription
Audio Intelligence
LeMUR
Pricing

Use Cases

Telephony Services

Learn

Documentation
Changelog
Tutorials
Industry News
Deep Learning
Engineering

Company

About
Careers
FAQs
Contact Us
Terms of Service
Privacy Policy
Subprocessors