ASC: Aggregating Sentence-level Classifications for Multi-label Long Text Classification

Takahiro Komamizu

Nagoya University, Japan

NAGOYA UNIVERSITY

Text Classification: the fundamental technology in NLP

- Sentiment Analysis ➔ Classify documents into sentiments (pos vs. neg)
  - Customer Feedback: Analyzing customer reviews, social media comments, and surveys to gauge public sentiment.
  - Brand Monitoring: Identifying positive or negative sentiments about a brand or product.
- Chatbots and Virtual Assistants ➔ Estimate the intension of users' claim
  - Customer Support: Automating responses to common queries in e-commerce, banking, and other sectors.
  - Personal Assistants: AI-driven systems like Siri, Alexa, and Google Assistant.
- Spam Detection ➔ Detect harmful content in a document
  - Email Filtering: Identifying and categorizing spam and phishing attempts.
  - Content Moderation: Detecting inappropriate or harmful text in forums or platforms.

etc.

Multi-label Long Text Classification (MLLTC)

- <u>Definition</u>

  Special case of text classification that

  - **Longer text** (than the length limit of classification models; esp. pre-trained language models (PLM))
  - **Multiple labels on the text** (imagine the fine-grained labels like topics of your papers)

- <u>Challenges</u>
  - Handling long text within PLM input length limits
  - Predicting multiple labels, especially for tail classes (caused by long-tail distribution)

ASC: Aggregating Sentence-level Classifications for Multi-label Long Text Classification

Takahiro Komamizu[0000−0002−3041−4330]

Nagoya University, Nagoya, Japan
taka-coma@acm.org

Abstract. Classification is a fundamental task for metadata estimation in archival document management within a digital library. Although pre-trained language models (PLMs) have evolved significantly, multi-label long text classification (MLLTC) remains challenging for PLM-based text classification methods due to their input text length limitations. Existing PLM-based classifiers typically utilize a single representation for a long text. In contrast, this paper explores a sentence-level classification approach. The basic idea is two-fold: a sentence in a text can often focus on one or a few classes, meaning multiple classes can be derived from the individual sentences; furthermore, sentences can typically fit within the length limit. There are two main issues with implementing a sentence-level classifier: the loss of context for each sentence and the increased training cost due to the larger number of documents that need to be processed by a PLM-based model. To address these issues, this paper proposes a framework, ASC, that uses sentence-level n-grams to form a sentence representation and employs a sentence selection method to reduce the number of sentences needed for training. The experimental results demonstrate that ASC outperforms existing text-level classifiers, achieving 25% and 48% improvements in Macro F1 metrics.

Keywords: Long Text Classification, Multi-label Classification, Prediction Aggregation, Sentence-level Classification

1 Introduction

Text classification [18, 14] has long attracted attention from the Digital Library (DL) community, with the expectation of supporting activities such as searching, browsing, recommending, and visualizing [7]. Text classification has advanced through mechanisms such as the Transformer [22]. The performance of text classification has been significantly enhanced by the successful implementation of pre-trained neural language models (PLMs), such as BERT [11] and RoBERTa [16], which are trained on vast amounts of text data to construct better text representations. Recent literature also considers token (or word) information as an important auxiliary factor for text classification [24, 15, 5]. BertGCN [15], a successor of TextGCN [24], constructs a token-wise graph and applies a graph convolutional network (GCN) to it, thereby obtaining token-oriented document representations.

Research Paper

Topic Labels
- Long Text Classification
- Multi-label Classification
- Prediction Aggregation
- Sentence-level Classification
- Pre-trained Language Models
- Extractive Summarization
- Sentence-level n-grams
- Class Imbalance
- Efficient Training

How to deal with lengthy documents?

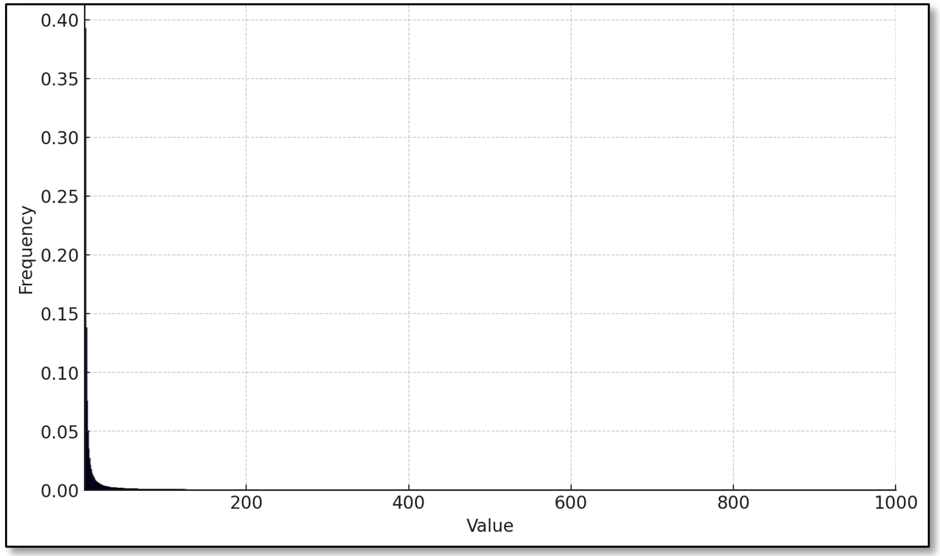
LLM (Large Language Model)?
➔ Let's discuss at the end.

- Approaches
  - Develop a model that can handle longer text (e.g., Longformer[2])
  - Decompose the document into segments (e.g., ToBERT[19])
- Experiments by Park et al.[20]: These approaches performed **comparably** with the simple methods (e.g., BERT[11])

Findings of Dai et al.[9]:

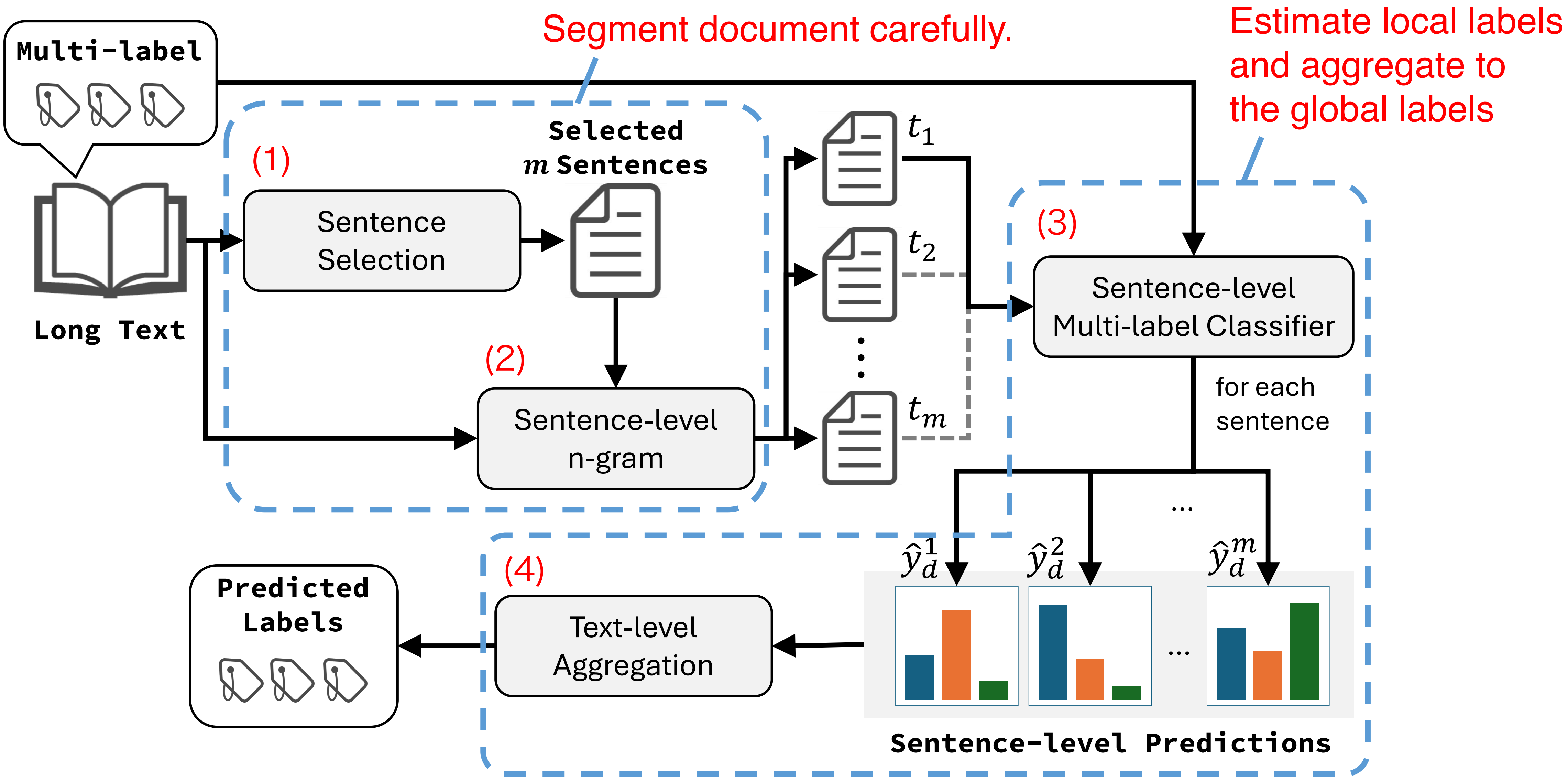
- *"Small local attention windows are effective and efficient"*
  ➔ **Segments should not be so large**.
- *"Splitting documents into overlapping segments can alleviate the context fragmentation problem."*
  ➔ Making **segments overlapped** to keep each segment contextually rich.

Why text classification methods suffer from multi-label nature?

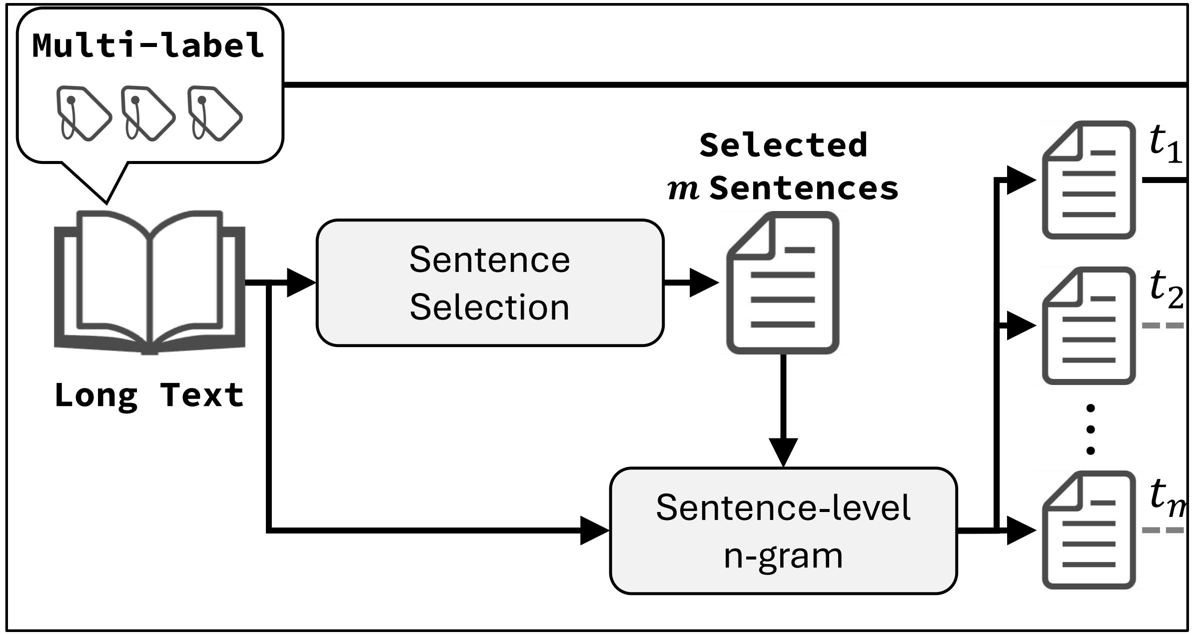


- Long-tail issue on the skewed distribution
  - Some labels appear very few in the corpus
    ➔ These few labels are not well trained by models (a.k.a. Class Imbalance).
    - e.g., Binary cross entropy (BCE) loss function maximizes accuracy.
- Every sentence is not always relevant to all labels associated to the text.
  - Some sentences (or paragraph) are related to a few labels.
  - In total, such labels for all the sentences compose the total set of labels of the text.

Proposed Simple-yet-Effective Framework: ASC



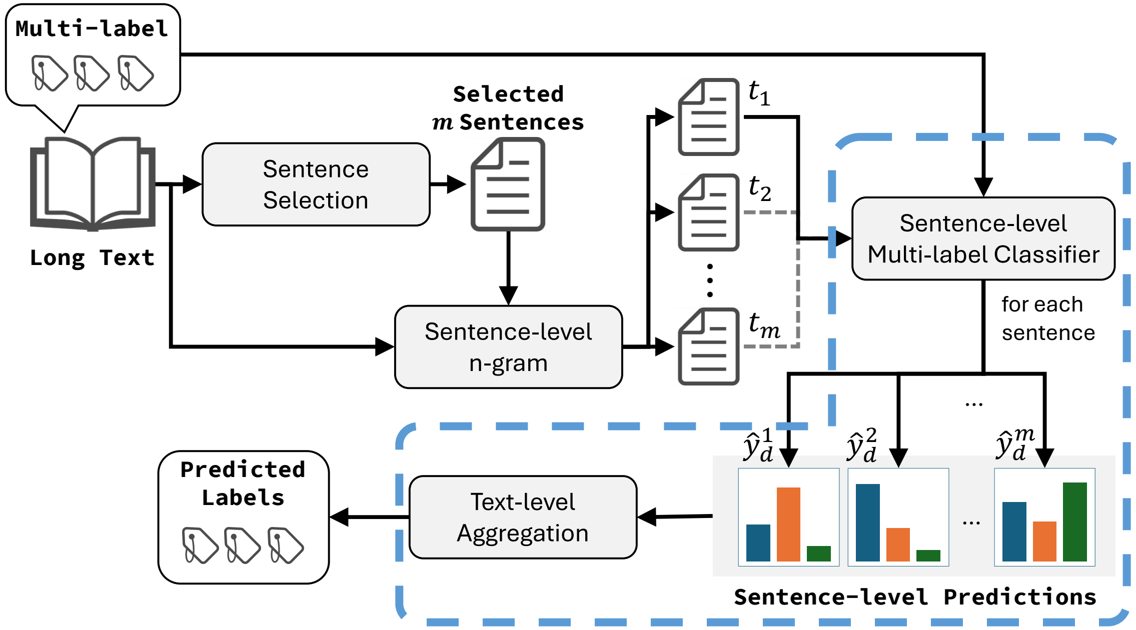
ASC - Sentence Selection



- Problem:
  - Large segment:
    Too many sentences introduce noise
  - Small segment:
    Losing contexts
    ➔ possible loss of proper meaning
- Solution: parameterization to control the segment
  - Extractive summarization (e.g., TextRank[17]) to select k key sentences
  - Sentence-level n-gram for context reconstruction

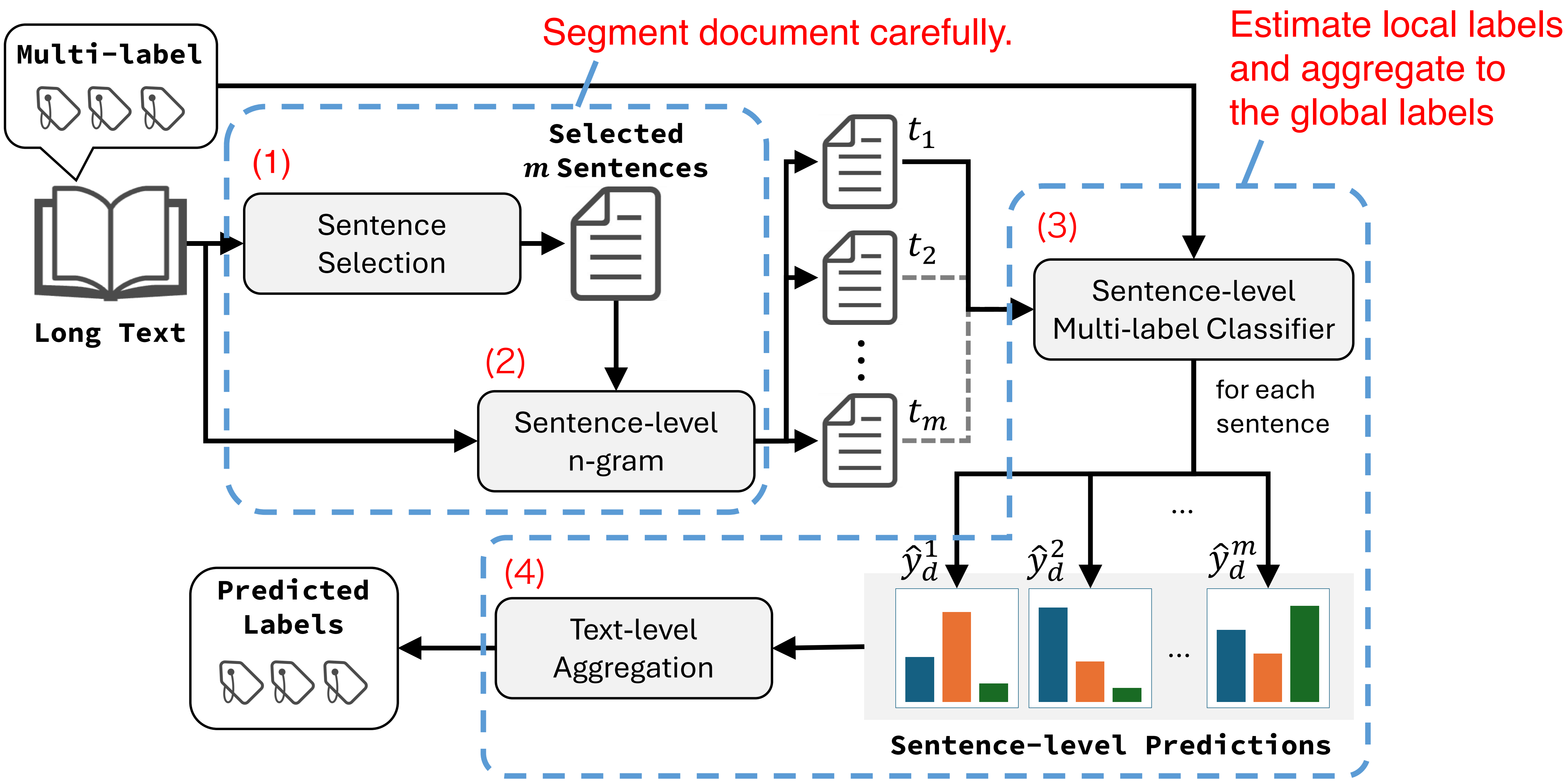
ASC – Sentence-level Classification & Text-level Aggregation

- Sentence-level Classification
  - In training, segments are associated with the text-level labels.
  - The model is expected to automatically recognize semantics between sentences and labels.



- Aggregation functions for text-level label estimation
  - **Mean**: segments may share similar labels, therefore, labels estimated for whole segments should be the text-level labels.
  - **Max**: segments may be exclusively related to labels, therefore, the combination of significant labels for each segment should be the text-level labels.

(Revisited) Proposed Simple-yet-Effective Framework: ASC



Datasets and Experimental Setup

- Datasets:
  - Reuters-21578
  - EURLex-57K

Name	D_{train}	D_{test}	L	$\overline{L}_T$	$\overline{S}_T$	$\overline{W}_S$
Reuter-21578	7,775	3,019	115	1.2	6.8	21.8
EURLex-57K	45,000	6,000	4,271	5.1	13.2	51.0

$\bar{L}_T$: avg. #labels / text
$\bar{S}_T$: avg. #sent. / text
$\bar{W}_S$: avg. #words / sent.

- Metrics:
  - Accuracy: a standard metric, but this can suffer from class imbalance.
    - Even if a model performs far better in the major classes than in the minor classes, this score can be higher.
  - Macro-averaged Precision, Recall, and F1 score
    - Class-wise averaging make robustness to the class imbalance.
- Baseline methods: DistilBERT variants, ToBERT, LongFormer

Experimental Results

Watch the gaps b/w comparison methods w/ ASC

Method	Reuter-21578						EURLex-57K					
	k	n	Acc ↑	Pre ↑	Rec ↑	F1 ↑	k	n	Acc ↑	Pre ↑	Rec ↑	F1 ↑
In-Length Limit Methods												
DistilBERT-head	-	-	.843	.382	.296	.322	-	-	.202	.171	.110	.127
DistilBERT-ES	5	-	.838	.382	.300	.325	5	-	.153	.153	.097	.111
DistilBERT-ES	10	-	.834	.383	.293	.319	10	-	.162	.158	.099	.114
DistilBERT-ES	20	-	.836	.389	.289	.317	20	-	.169	.162	.101	.117
Long Text Handling Methods												
DistilBERT-Rand [20]	-	-	.858	.497	.386	.419	-	-	.224	.221	.163	.179
DistilBERT-TR [20]	-	-	**.860**	.488	.366	.397	-	-	.225	.230	.167	.184
ToBERT [19]	-	-	.850	.478	.377	.406	-	-	.166	.137	.071	.087
LongFormer [2]	-	-	.850	.450	.359	.384	-	-	.099	.131	.086	.099
Proposed Methods												
ASC-mean	9	11	.852	**.548**	.457	.486	8	9	**.274**	**.310**	.247	.263
ASC-mean w/o ES	-	11	.844	.546	.453	.480	-	9	.269	**.310**	.251	.267
ASC-max	9	11	.824	.539	**.537**	**.524**	8	9	.236	.262	**.318**	**.273**
ASC-max w/o ES	-	11	.819	.522	.530	.516	-	9	.240	.261	.309	.270

Sentences are sampled.

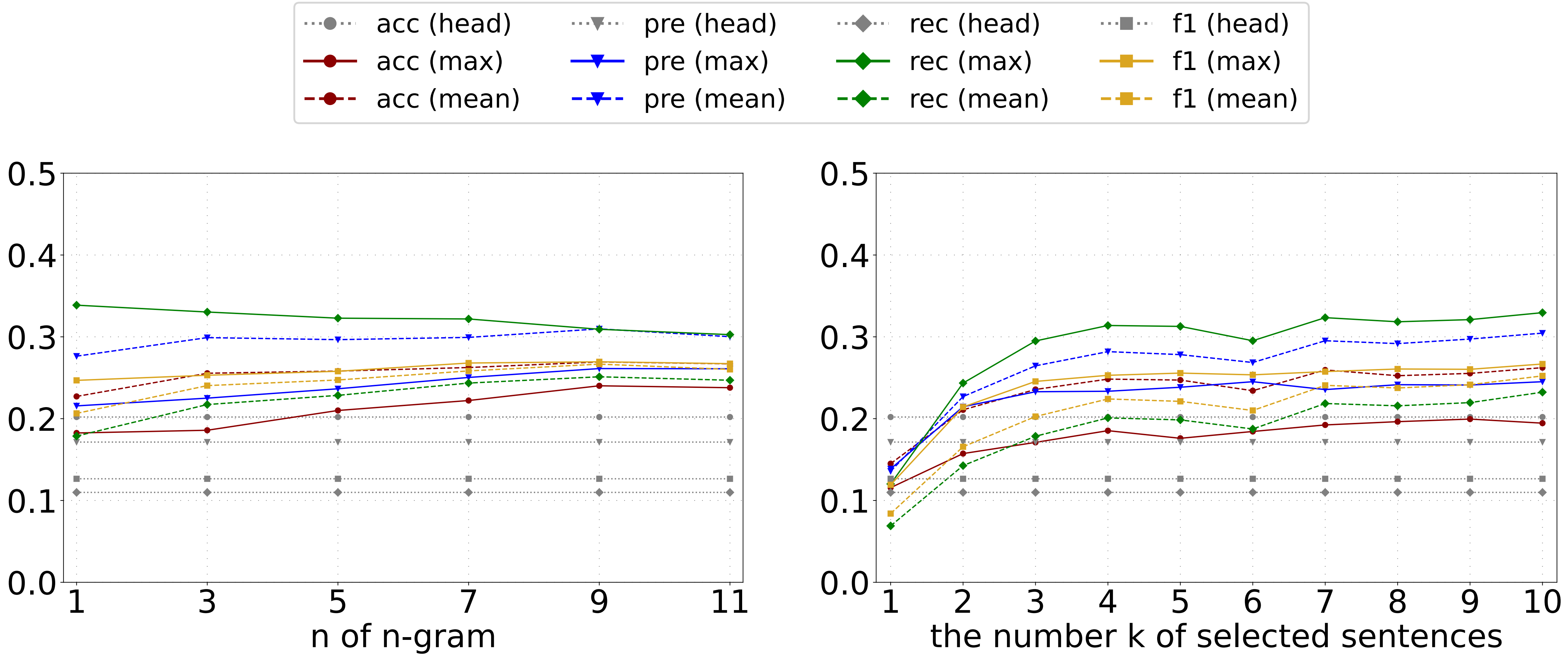
Head sent. plus sampled sent.

Small gap in Acc.
➔ Estimation to the similar number of texts are correctly

High gap in F1.
➔ Estimation to the minor classes is more correct.

Higher Rec.
➔ Estimation to the minor classes is more aggressive.

Insights and Sensitivity Analysis



Conclusion

- Summary of contributions:
  - Novel sentence-level approach for MLLTC
  - Effective handling of context and noise
  - ASC as a promising framework for future NLP applications
- Strengths of ASC
  - Handles long texts efficiently
  - Improves prediction for tail classes
  - Robust to context loss via n-grams
- Limitations and Future Work
  - Training cost with large datasets
  - Potential for advanced aggregation methods
  - Addressing class imbalance in extreme scenarios