



INDIAN LANGUAGE RESOURCES— TEXT PROCESSING SUBCOMMITTEE REPORT

Authored by

IEEE SA Text Processing Subcommittee

TRADEMARKS AND DISCLAIMERS

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

The ideas and proposals in this specification are the respective author's views and do not represent the views of the affiliated organization.

ACKNOWLEDGMENTS

Special thanks are given to the following reviewers of this paper:

Amitava Das

Girish Nath Jha

Niladri Sekhar Dash

The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.

All rights reserved. 18 September 2023. Printed in the United States of America.

PDF: STDVA26428 979-8-8557-0067-1

IEEE is a registered trademark in the U. S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated. All other trademarks are the property of the respective trademark owners.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system, or otherwise, without the prior written permission of the publisher.

Find IEEE standards and standards-related product listings at: <http://standards.ieee.org>.

NOTICE AND DISCLAIMER OF LIABILITY CONCERNING THE USE OF IEEE SA INDUSTRY CONNECTIONS DOCUMENTS

This IEEE Standards Association (“IEEE SA”) Industry Connections publication (“Work”) is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the IEEE SA Industry Connections activity that produced this Work. IEEE and the IEEE SA Industry Connections activity members expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the IEEE SA Industry Connections activity members disclaim any and all conditions relating to: results; and workmanlike effort. This IEEE SA Industry Connections document is supplied “AS IS” and “WITH ALL FAULTS.”

Although the IEEE SA Industry Connections activity members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE OR IEEE SA INDUSTRY CONNECTIONS ACTIVITY MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patents claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder. The policies and procedures under which this document was created can be viewed at <http://standards.ieee.org/about/sasb/iccom/>.

This Work is published with the understanding that IEEE and the ICom members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

TABLE OF CONTENTS

1. INTRODUCTION	6
2. SCOPE	7
3. TEXT PROCESSING USE CASES	8
3.1. CHARACTER MODEL USE CASES	8
3.1.1. TEXT VARIATION IN SYNTACTIC CONTENT UNDER HTML/CSS APPLICATION.....	8
3.1.2. INDIAN LANGUAGE SEARCH OPERATIONS ON THE WEB	10
3.1.3. COUNTER STYLE APPLICATIONS.....	10
3.2. WORD-LEVEL PROCESSING USE CASES.....	12
3.3. SENTENCE-LEVEL PROCESSING USE CASES.....	13
3.4. USE CASES FOR DISCOURSE PROCESSING	14
3.5. USE CASES FOR CODE-MIXING	15
3.6. USE CASES FOR DOWNSTREAM NLP TASKS.....	16
4. DIFFERENT (SUB)TASKS FOR TEXT PROCESSING.....	17
5. SURVEY OF CURRENT STANDARDS	18
5.1. STANDARDS AVAILABLE FOR THE CHARACTER LEVEL TEXT PROCESSING	19
5.2. STANDARDS AVAILABLE FOR THE WORD-LEVEL TEXT PROCESSING	20
5.3. STANDARDS AVAILABLE FOR THE SENTENCE-LEVEL TEXT PROCESSING	22
5.4. STANDARDS AVAILABLE FOR THE DISCOURSE PROCESSING	24
5.5. STANDARDS AVAILABLE FOR THE CODE-MIXED INPUT.....	25
6. GAPS IN STANDARDS FOR INDIAN LANGUAGES.....	26
6.1. GAPS IN CHARACTER-LEVEL TEXT PROCESSING	27
6.2. GAPS IN WORD-LEVEL TEXT PROCESSING	27
6.3. GAPS IN SENTENCE-LEVEL TEXT PROCESSING	28
6.4. GAPS IN DISCOURSE-LEVEL TEXT PROCESSING	29
6.5. GAPS IN DOWNSTREAM TASKS	29
6.5.1. CASE STUDY 1: GAPS IN QUESTION ANSWERING	30
6.5.2. CASE STUDY 2: GAPS IN SENTIMENT ANALYSIS.....	30
7. TEXT PROCESSING RESOURCES.....	30
7.1. DATA AND TOOLS	31
7.1.1. WORD LEVEL.....	31
7.1.2. SENTENCE LEVEL.....	31
7.1.3. DOWNSTREAM APPLICATION.....	32

TABLE OF CONTENTS

7.2. DOCUMENTS/PAPERS ON PREVIOUS STANDARDS.....	33
7.2.1. CHARACTER LEVEL	33
7.2.2. WORD LEVEL.....	33
7.2.3. SENTENCE LEVEL.....	34
8. REFERENCES.....	34

TEXT PROCESSING SUBCOMMITTEE REPORT

1. INTRODUCTION

A large portion of language processing considers text as the input. This text input varies from task to task, language to language. Every task operates on some granularity of the text as input (ranging from as small as a character to as large as document collection) and produces the desired output. The output essentially conveys the underlying information for different decision-making in a product or helps other language-processing tasks. Thus, maintaining standards in both input and output helps developing a unified information representation, which subsequently helps the NLP pipeline and software.

A lot of research has been done on different NLP tasks and standards both internationally and in Indian languages (ILs). Much software has been built around these tasks and widely used in products. However, often different research and product groups have created different standards to address the problem. This often causes issues in sharing of data, information representation, etc. In this report, the authors investigate the NLP text processing tasks for which standardization is required and subsequently explore the different standards available either in ILs or internationally. They categorize the tasks primarily based on their input/output for this study. Furthermore, they also conduct few case studies based on downstream applications.

The primary objective of this study (focusing ILs) is as follows:

- Identify the text processing tasks where standardization is required.
- Collect different standards available for different text processing tasks.
- Gaps and future needs.

The rest of the document is organized as follows. In Section 2, the scope of this study is described. Section 3 provides the list of tasks considered in this study which potentially need standardization. In Section 4, different use cases for each task are provided. A detailed survey of the available standards is described in Section 5. Section 6 focuses on the gaps in the current standards. Finally, all the resources (both tools and data) are listed in Section 7, which provides a consolidated view of the resources available for future analysis and study.

2. SCOPE

The scope of this survey includes the following:

1. There are quite many text processing tasks that exist in the field of NLP. However, while some tasks require standards [like POS tagset, named entity recognition (NER), etc.], some may not need the same (word sense, domain terms, etc.). Identify the tasks for which standardization is required. This includes the standardization in the annotation (including tagset and guideline), definition, and formatting for the different text processing tasks. In this survey, the authors categorize the tasks based on the input in the following dimensions and explore different tasks in each dimension and the need for standardization:
 - a. Character-level
 - b. Word-level
 - c. Sentence-level
 - d. Discourse-level
 - e. Code-mixed

In addition to the above, the survey also includes a few end-user applications (e.g., question answering (QnA), summarization) as case studies to understand the need for standardization for specific tasks.

2. A detailed survey of the existing standards available primarily focusing ILs. The survey also includes global standards whenever available to understand the big picture across international languages.
3. Furthermore, the scope of the survey also includes documenting available resources and tools.
4. Identifying the gaps.

The scope of the task does not include the following:

1. There are many approaches to solve each text processing task. Some of them may have some correlation with the granularity of annotation (e.g., number of tags). The survey does not include any study on approaches used to solve the tasks.
2. Often there are multiple standards (originated from different research groups at the same time) available for a task. This survey lists all of them and does not compare them side-by-side as the same is not the scope of the prestandardization. Rather, generalize the gaps across them.

3. TEXT PROCESSING USE CASES

In the previous section, the authors mentioned different levels of text processing based on the smallest input unit used in the processing. In this section, they list potential use cases for each of these tasks. They describe the use cases of character, word, sentence, and discourse-level text processing from Section 3.1 to Section 3.4, respectively. Furthermore, they also discuss the use cases of different downstream tasks in Section 3.5.

3.1. CHARACTER MODEL USE CASES

The W3C character model for the world wide web: String matching provides authors of specifications, software developers, and content developers a common reference on string identity matching on the world wide web. The goal of this document is to enable the web to process and transmit the text in a well-defined and well-understood way.

W3C string searching document covers string searching operations on the web in order to allow greater interoperability. String searching refers to natural language string matching such as the “find” command in a web browser.

The web is primarily made up of document formats and protocols based on character data. These formats or protocols can be viewed as a set of resources consisting mainly of text files that include some form of structural markup or syntactic content. Processing such syntactic content or document data requires string-based operations such as matching (including regular expressions), indexing, searching, sorting, and so forth. This document examines different types of text variation that affect both user perception of text on the web and the string processing on which the web relies. The W3C character model document defines one of the basic building blocks for the web related to this problem by defining rules and processes for string identity matching in document formats. These rules are designed for the identifiers and structural markup (syntactic content) used in document formats to ensure consistent processing.

3.1.1. TEXT VARIATION IN SYNTACTIC CONTENT UNDER HTML/CSS APPLICATION

Syntactic content is any text in a document format or protocol that belongs to the structure of the format or protocol. The element name, ids, and class name and various attributes are part of syntactic content in HTML/cascading styles sheet (CSS). When producing content, you should ensure that selectors and class or id

names are character-for-character the same. Consider the below example:

```
<!DOCTYPE html>
<html>
<head>
<style>
#हिंदी-शैली-ज़ांच {
    text-align: center;
    color: red;
}
</style>
</head>
<body>
<p id=" हिन्दी-शैली-ज़ांच">Text in red color</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

In the above example, the id name defined in the HTML and CSS uses the same character sequences. There will be gaps if id name uses different character sequences. This is particularly likely to be an issue if the markup and the CSS are being authored or maintained by different people.

हिंदी /hīdi/= 0939+ 093F+0902+0926+0940
हिन्दी /hīdi/= 0939+ 093F+0928+094D+0926+0940
ज़ांच /zatj/= 095B+093E+0902+091A
ज़ाच /zatj/= 091C+093C+093E+0902+091A

Resources are often susceptible to the effects of these variations because their specifications and implementations on the web do not require Unicode normalization of the text, nor do they take into consideration the string-matching algorithms that are used while processing the syntactic content and natural language content.

3.1.2. INDIAN LANGUAGE SEARCH OPERATIONS ON THE WEB

Users can search natural language content by using the find command on the web. Different character sequences of the same text might be used by different users by performing the find command. When searching or matching text by comparing code points, these variations in encoding cause text values not to match that users expect to be the same. A few examples are shown below:

आवाज़ /avaz/ :0906 + 0935 + 093E + 095B

आवाज़ /avaz/ :0906 + 0935 + 093E + 091C + 093C

फ़ाँसी /fāsi/ :095E + 093E + 0901 + 0938 + 0908

फ़ाँसी /fāsi/ :092B + 093C + 093E + 0901 + 0938 + 0908

Additionally, in ILs some of the texts represent two orthographic representations with different encodings. The spelling variations lead to introducing the inappropriate searching results. Some examples are shown below:

हिंदी /hīdi/ & हिन्दी /hīdi/, मंडी /māḍi/ & मण्डी /māḍi/, चम्पक /tjāpək/ & चंपक /tjāpək/, ठंडा /tʰāḍa/ & ठण्डा /tʰāḍa/,

अम्वर /āṃvər/ & अंवर /āṃvər/

ALL ILs/script characters need to be postprocessed via normalization for correct IL string manipulation on the web.

3.1.3. COUNTER STYLE APPLICATIONS

W3C CSS counter styles (listing) level 3 specification (<https://www.w3.org/TR/css-counter-styles-3/>) provides a mechanism for authors to create user-defined counter styles (ordered/unordered listing) with the help of @counter-style rule, which allows authors to define their own custom counter styles for use with CSS list-marker and generated-content counters (listing). The numbering styles of English, Hindi, and other ILs have already been implemented and supported by various browsers. The alphabetic listing/bulleting has not been implemented by various browsers, and the same is not currently reflected in the specification as shown in this section (<https://drafts.csswg.org/css-counter-styles/#simple-alphabetic>). For this purpose, W3C has implemented readymade counter styles (<https://w3c.github.io/predefined-counter-styles/>) that can be used as a reference for those who wish to create their own user-defined counter styles for CSS.

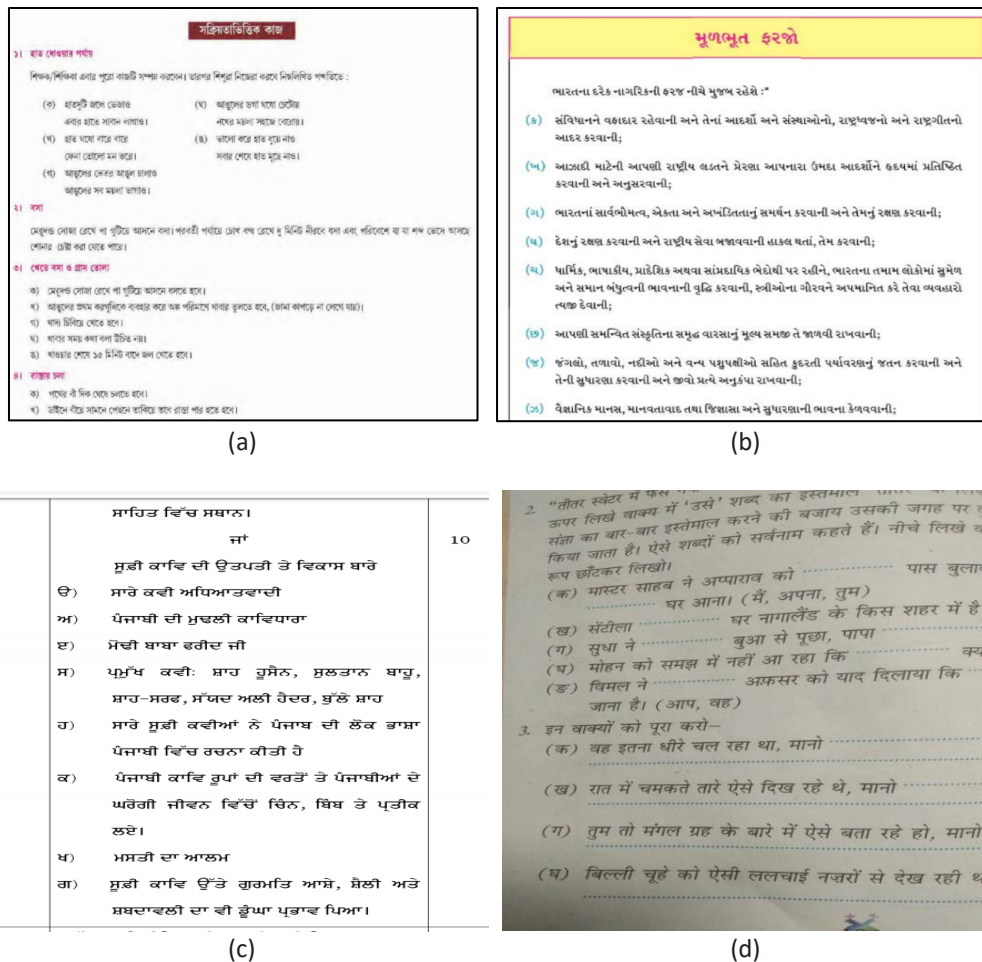
These counter styles will be used by various cultures around the world and can be used as a reference for

those wishing to create their own user-defined counter styles for CSS style sheets.

- ILs alphabetic/numeric listing in web interface through CSS.
- These listings will also provide a mechanism for developers to define ILs listing in the digital publishing domain. Currently, the counters are mainly used in textbooks, magazines, newspapers, etc. Some of the following snapshots show the usage of alphabetic listing in ILs.

FIGURE 1 provides some examples of different counter styles in different languages.

FIGURE 1 Counter style used in different languages. (a) Bengali. (b) Gujarati. (c) Punjabi. (d) Hindi



3.2. WORD-LEVEL PROCESSING USE CASES

The authors emphasize the several use cases of different linguistic information available through word-level processing. They primarily focus on the nine different word-level processing tasks and their use cases for different downstream NLP applications.

- a. Tokenization is the process of separating punctuations from the input text. This is the first step toward any text processing and is required for all downstream applications. This is also required for deep-learning (DL)-based solutions.
- b. Segmentation (word/sentences) is the process of dividing written texts into smaller meaningful units. For example, sentence segmentation divides a text into its component sentences. Similarly, a word segmenter generates the component words of an input text. This is also an integral part of tokenization, especially when the input is longer than a sentence or does not have a clear word boundary identifier.
- c. Morphological attributes depict all the information available for a word. This is desirable information for many NLP tasks (including POS, NER, MT, etc.) to achieve better results (especially for morphologically rich ILs). However, this is often required while solving NLP problems using a reductionist approach.
- d. POS tagging assigns a unique grammatical category to each token in an input. This is required for the development of many NLP modules (Parser, Chunker, etc.) and downstream applications like TTS, NER, WSD, etc.
- e. NER identifies different names (often includes person, location, and organization) available in a text. This is required for many downstream applications including translation, search engine, speech, intent classification, etc. This is often important irrespective of whether a downstream model uses a classical pipelined solution or a DL-based solution for the downstream tasks.
- f. Multiword expression (MWE) extracts the idiosyncratic unit that crosses word boundaries. This is also required for many applications (like NER) where MWEs need to be treated separately as their compositional meaning differs from the individual meaning of the associated words.
- g. Chunks are the group of words that act as a constituent, basic unit for deep parsing.
- h. Transliteration is the process of converting a word from one language to another language alphabetically. This has wide applications in machine translation, search engines, CRM software, etc.

Note that due to the advancement in DL-based techniques, many applications (especially for high-resource languages) currently prefer to use end-to-end solutions instead of pipelined architecture. The DL-based techniques have reduced the scope of a few of the aforementioned NLP tasks [e.g., (c), (d), and (g)]. However, the use cases for (a), (b), (e), (f), and (h) are still very wide in NLP applications.

3.3. SENTENCE-LEVEL PROCESSING USE CASES

- a. Dependency parsing:
 - 1. Dependency tags can be used as features for many DL and machine learning modules to learn intent, sentiment, emotion, etc.
 - 2. Dependency tags can play a good role in natural language understanding systems such as conversational agents (human–machine interaction) as well as understanding conversation between two humans. For example, understanding conversations between two users to automatically create events, reminders, etc.
 - 3. Dependency tags can help in developing chatbots too.
- b. Phrase-structure parsing:
 - 1. This shows chunking information too.
 - 2. These tags can also be used as features for many ML/DL models.
- c. Sentiment annotation:
 - 1. To automatically understand the reviews of a product on the online platform and help in good recommendations.
 - 2. The same will be applicable to movie reviews, book reviews, etc., and can enrich all recommendation systems.
- d. Hate speech detection: Hate speech is defined as “any communication that disparages a person or a group based on some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic.”
- e. Emotion detection: Identifying the emotion of a sentence is becoming important in many use cases.
 - 1. Understanding the emotion of user’s review for any product.
 - 2. By understanding the emotion of a sentence, the authors can recommend appropriate emoticons.