

A Novel Sentence Segmentation Approach For Test Cases Using NLP

A White paper using NLP Solution

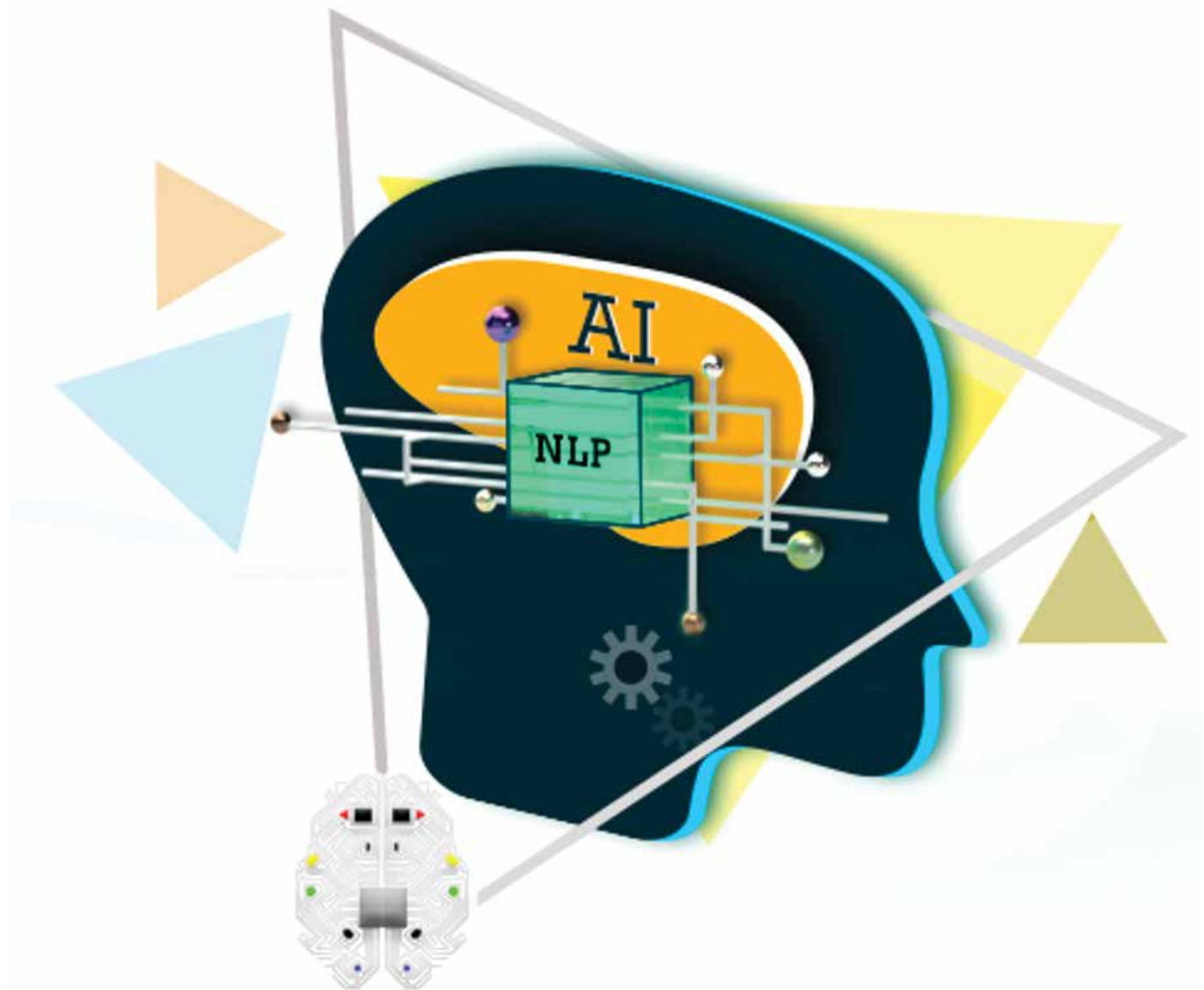


TABLE OF CONTENTS

Abstract	3
Introduction	4
Target Audience	6
Solution	7
Benefits	14
Conclusion	15
Reference	15
Author Info	16
Sujith Viswanathan	16
Srihari V	16

Abstract

Software Testing Life Cycle (STLC) defines a sequence of activities to ensure the quality of software. The Test Case development phase in STLC helps in developing the test cases documents based on requirements and standards. A Test case consists of a sequence of test steps that validates whether a condition being tested on application satisfies its requirements.

At times, test steps are written in a way (complex or compounded sentences) that is difficult to process directly for Artificial Intelligence-based data or keyword-driven automation frameworks.

The latest researches in the area of Artificial Intelligence conveys a high success rate in the usage of Machine Learning (ML) Algorithms for Natural Language Processing tasks. NLP techniques provide systems the intelligence to understand and perform various actions on human-understandable languages. The test steps from test case documents have to be processed and provided in a format, which can be understood by Automation frameworks that use ML algorithms for further processing.

This proposed solution will act as an intermediate tool. It will assist in breaking down test steps into simpler meaningful sentences. This can be done by maintaining the correct order of precedence and semantic structure. In this approach, processing test steps becomes easier for AI-driven frameworks that process test steps into the keyword or data-driven formats. This is achieved through a sentence segmentation model that uses Neural Networks. The segmentation handles both prepositional and conjunction-based sentences.

This solution can be integrated into any AI-based automation framework for ease of handling the test steps. This solution can be used in assisting a majority of the existing ML algorithms to detect and get Keyword output in the desired format.

Introduction

Testing is one of the most important phases of the software life cycle [1]. Testing is either done manually or through automation. A large number of frameworks are available in the market to perform automation testing [2]. The process of providing systems the ability to interact with texts given by humans can be considered as one of the most improving areas of Artificial Intelligence (AI).

This encouraged to focus on innovation of Artificial Intelligence techniques that can make automation frameworks intelligent. The proposed sentence segmentation approach makes use of Machine Learning (ML) and Deep Learning algorithms for getting meaningful phrases as segments [3]. Machine Learning is an AI technique that provides systems the intelligence to perform complex tasks that humans perform.

The area in AI that provides interaction between systems and texts is referred to as Natural Language Processing (NLP) [4]. The two major domains under NLP are Natural Language Understanding (NLU) and Natural Language Generation (NLG) [5]. NLU deals with the extraction of information from a text. NLG deals with processes that can make a system write or report in the form of texts. NLP tasks can play a huge role in the semantic understanding of every test step. This provides the flexibility for developers to write test cases of their interest.

The NLP techniques can segment the test steps in test case documents. Sentence segmentation is a demanding task in NLP, as a sentence (test step) has to be split into segments based on their semantic structure [6]. The sentence segmentation approach discussed in this paper is an innovation that is not restricted to the testing domain alone. The segmentation is currently implemented on conjunctions and prepositions like “and”, “from”, and “under”. The same approach can be extended to perform the segmentation of any other conjunction or preposition.

A majority of the engineers tend to use conjunction (“and”) and preposition (“from”, “under”, etc.) while writing test steps to combine multiple actions in one test step. This brings in complexity for the frameworks to understand and extract the required keywords.

Whenever AI/ML-driven framework detects a sentence consisting of conjunction/preposition during test case document processing, the framework faces a challenge to extract keywords/actions from the test step. In such scenarios, the sentences need to be segmented to take the right keywords out of it. In most cases, the framework is designed to expect one keyword for a test step. Therefore, the sentence segmentation approach plays a key role in breaking down the sentences and extracting meaningful phrases.

For a test case document provided as input, the segmentation approach accomplishes the following:

1. Checks the possibility to find meaningful phrases from an input test step.
2. Splits test step while encountering a conjunction or preposition.
3. Takes care of the order and meaning of every segmented output.

The above-mentioned stages make this AI-based solution more intelligent and less complex when compared to other similar innovations [7].

Target Audience

In the market, a majority of the BDD-based frameworks operate based on keywords to complete the desired task. In such frameworks, the observation is that a user should be aware of every keyword developed or available in it. Apart from this, the user may need to provide the predefined test step to the framework for extracting keywords. AI-based frameworks provide more intelligence to the framework components due to their ability to learn the necessary information.

The solution can be used by Behavioral-Driven Development (BDD), and AI-based Framework developers/inventors, to overcome the challenges in handling complex, compound, and non-meaningful (unstructured) sentences. Generally, test case documents comprise of such sentences as test steps.

For Example - FALCON is an end to end AI based testing solution [2], that uses the TAROT engine to extract the keywords directly from test steps present in the test case documents to perform the desired task. FALCON is a tool that doesn't generate any scripts for test executions.

The solution discussed here is integrated with the TAROT Engine [8]. It plays a major role in converting complex and compound sentences into a simplified form. The solution is domain-independent and can be incorporated into different applications.

The success of the solution in the test automation framework (FALCON) indicates its wide scope in Action, Keyword, and Behavioral Driven test automation frameworks. The features available in this solution makes it efficient to work as an important preprocessing technique for various AI based use cases like Question Answering (Chatbots), Speech Recognition (Speech to Text Engines), Machine Comprehension, Machine Translation, Fake News Detection, Sentiment Analysis, etc.

Solution

Data (test cases) generally includes inconsistency, noise, or missing values as it comes from different sources. The presence of incomplete data that has missing attributes, values, or lack of certain attributes might result in poor decision making for an AI solution. It is necessary to perform some modifications to such data for getting qualitative results.

Below are the few challenges observed with respect to processing test cases in AI frameworks.

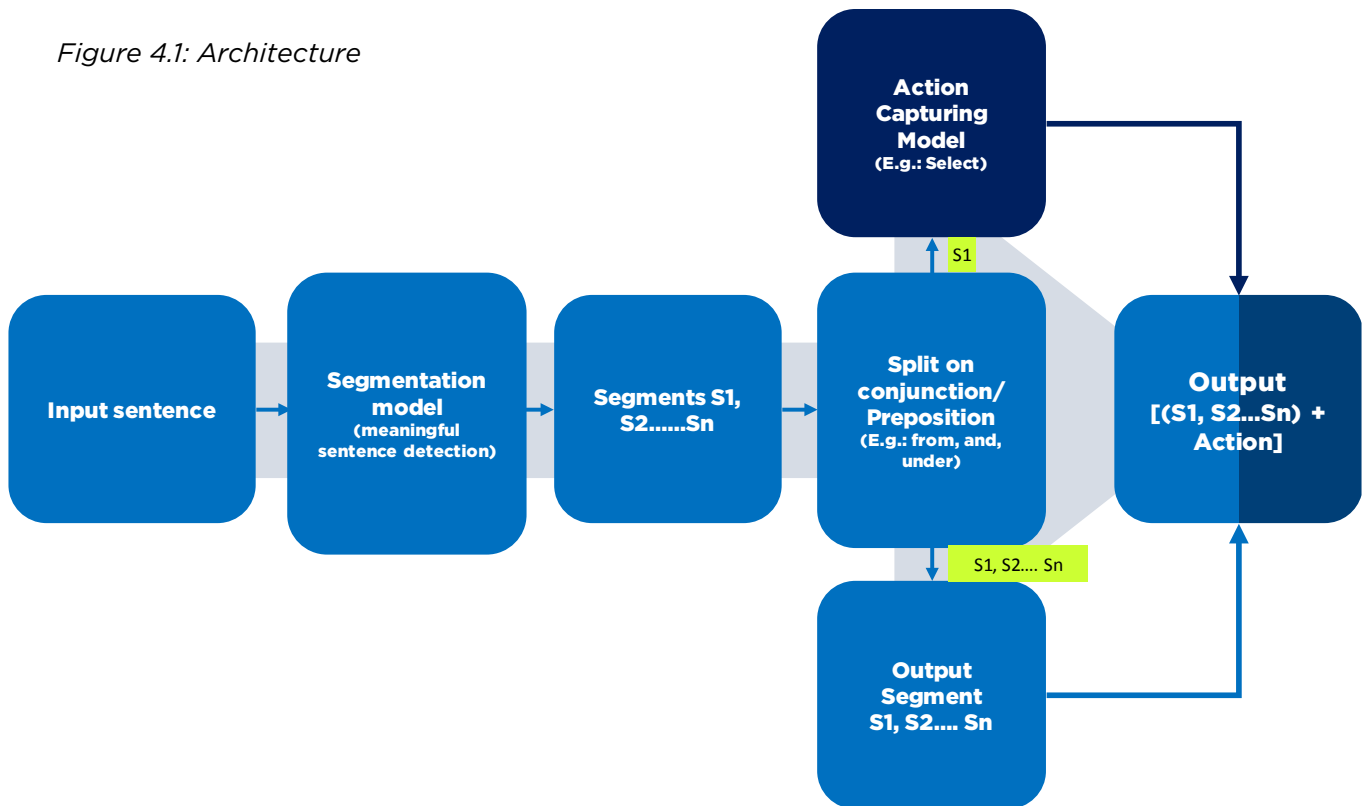
The image displays four light blue rectangular boxes arranged horizontally, each containing a numbered challenge. The numbers 1, 2, 3, and 4 are inside white circles at the top of each box. The text in each box is centered and bolded.

- 1** Test cases documents may not have granular details on test steps
- 2** In a few cases, the test step demands complexity in writing to meet requirements
- 3** The English language restricts the engineers while developing test cases to write test steps in a certain style which differ from the actual order of execution
- 4** The presence of conjunction or preposition in a sentence (test step) may require action to be transferred to all segmented phrases.

The test case document will be the input for the tool [8]. The stage where the data modification process occurs can be referred to as the data preprocessing stage. The solution primarily aims at the preprocessing stage.

The Segmentation Engine checks for the presence of any conjunction or preposition ('from', 'and', 'under' or any similar) in a sentence. While encountering a conjunction or preposition, every test step present in the form of a sentence can be segmented into two or more meaningful phrases. An example of the segmentation solution on "from" is shown in Table 4.1

Figure 4.1: Architecture



The Segmentation Engine of the architecture takes into consideration that all the segmented outputs should maintain the order of precedence and semantic meaning. In order to achieve the requirement, this module uses an action capturing model (ML-based) built on our corpus to detect the actions on every sentence so that every segmented output is merged with required action to make them meaningful.

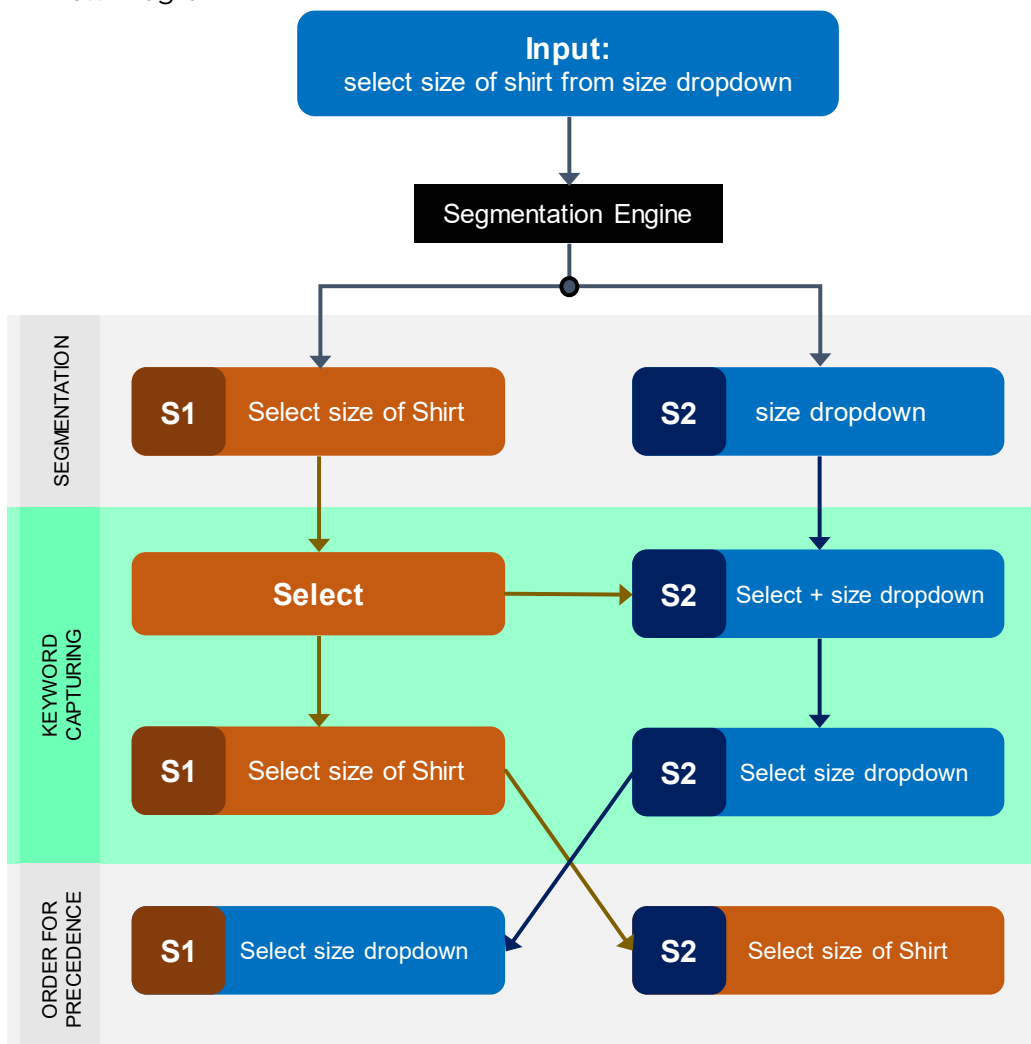
Case 1: Punctuated (Meaningful/Structured sentence)

Table 4.1: Scenario for Preposition

Input Test Case Document		Segmented Test Case Document	
Sr. No.	Test step	Sr. No.	Test step
1	select size of shirt from size dropdown	1.1	select size dropdown
		1.2	Select size of shirt

The above example (Table 4.1) represents a test step with a preposition (“from”). Here, the solution splits the sentence into two phrases. It is important to transfer the necessary action (“Select”) to segmented phrases. Then, the logic for the order of precedence will be taken care of by reordering the segmented phrases (Refer Figure 4.2 below to get a detailed understanding of the segmentation process).

Figure 4.2: Flow Diagram



Case 2: No Punctuation (unstructured sentences)

In addition, this segmentation solution with the Deepsegment model also identifies the segments in a sentence even if they are not separated by punctuation.

Table 4.2: Scenario for unstructured input sentence

Input Test Case Document		Segmented Test Case Document	
Sr. No.	Test step	Sr. No.	Test step
1	select size of shirt from size dropdown Click on 'Policies'	1.1	select size of shirt from size dropdown
		1.2	Click on 'Policies'

In Table 4.2, the input test case document is not a meaningful sentence. However, our model has the efficiency to split the non-meaningful sentence to possible meaningful sub-sentences [Segmented Test Case Document].

Case 3:

Table 4.3: Normal -Input Sheet Format

Test Case Document			
Sr. No.	Test step	Test Data	Expected Result
1	Select size dropdown		Size dropdown is selected
2	Select "size" of shirt	42	Size of shirt is selected
3	Click on "profile"	P1	Profile is clicked
4	Click on Orders		Orders is clicked
5	Click the "IN PROGRESS" status	A1	IN PROGRESS status is clicked
6	Click on any service request		Any service request is clicked

Normal Way of Writing

The below test step is a combination of all test steps in Table 4.3, This example highlights that the segmentation approach will have the intelligence to understand the steps and arrange in order of precedence.

1. The Test Step 1 and 2 in Table 4.3 can be simplified as, Select size of "shirt" from size dropdown. Similarly, 3 and 4 as, click on profile and click on Orders. 5 and 6 as, Click on any service request under the "IN PROGRESS" status.
2. With the help of this module, a user can write test steps with multiple actions given as shown in Table 4.4

Table 4.4 : Segmentation Module - Input Sheet Format

Input Test Case Document			
Sr. No.	Test step	Test Data	Expected Result
1	select size of "shirt" from size dropdown	42	Size dropdown is selected Size of shirt is selected
2	Click on profile and click on Orders	P1	Profile is clicked Orders is clicked
3	Click on any service request under the "IN PROGRESS" status	A1	IN PROGRESS status is clicked Any service request is clicked

Segmentation Way of writing

Engineers have the provision to write test steps with more than one conjunction or preposition in a sentence. The below shown examples [Table 4.5] depict how they are handled by the solution with the help of "**from**", "**under**", "**and**".

Table 4.5: Segmentation Model Outputs

Table 4.5.1: Case 1:from

	Input Test Step	Initial Segmentation	Final Segmentation (Order of precedence)
1.1	Select the option "BusinessRole" from the "SelectRole" dropdown	Select the option "BusinessRole"	Select the "SelectRole" dropdown
		Select the "SelectRole" dropdown	Select the option "BusinessRole"
1.2	Select from account from "from account" drop down list	Select account	Select "from account" drop down list
		Select "from account" drop down list	Select account
1.3	select operator from "operator" Dropdown	Select "operator" Dropdown	select "operator" Dropdown

Table 4.5.2: Case 2:and

	Input Test Step	Initial Segmentation	Final Segmentation (Order of precedence)
1.1	Click on profile and click on Orders	Click on profile	Click on profile
		click on Orders	click on Orders
1.2	Enter OTP and click on Continue payment	Enter OTP	Enter OTP
		Click on Continue payment	Click on Continue payment
1.3	Check the format of Checkout date and format	Check the format of Checkout date and format	Check the format of Checkout date and format

Table 4.5.3: Case 3:under

	Input Test Step	Initial Segmentation	Final Segmentation (Order of precedence)
1.1	Click on any service request under the "IN PROGRESS" status	Click on any service request	Click the "IN PROGRESS" status
		Click the "IN PROGRESS" status	Click on any service request
1.2	Check the data displaying under the dashboards pane	Check the data displaying	Check the dashboards pane
		Check the dashboards pane	Check the data displaying
1.3	Click on "Edit" button on service request under any status except "Review" from the Dashboard	Click on "Edit" button on service request	Click any status except "Review" from the Dashboard
		Click any status except "Review" from the Dashboard	Click on "Edit" button on service request

Table 4.6: Feature List

Features	DeepSegment	Spacy	NLTK Punkt	Segmentation Engine
Punctuation	✓	✓	✓	✓
No Punctuation	✓	✓	✓	✓
Correct Punctuation	✓	✓	✓	✓
Conjunction	✗	✗	✗	✓
Preposition	✗	✗	✗	✓
Action Capturing	✗	✗	✗	✓

The table [Table 4.6] represents the feature list of the segmentation solution in handling conjunctions and prepositions. It is observed that our solution's features are improved from the reference model (DeepSegment). When the reference model focused on No punctuation, our solution was built on top of it which handles both No punctuation and primarily Conjunction/Preposition based sentence segmentation.

Benefits

1

The solution allows Engineers to include multiple information in a test step by combining multiple actions using a conjunction/preposition.

2

The solution reduces the complexity in processing test case documents for AI/ML-based test automation frameworks.

3

The engineers are provided ease of handling test steps during keyword extraction.

4

Complex test steps (with multiple conjunction or prepositions) will be broken down into simple phrases.

5

The solution stands different from other implementations of sentence segmentation.

6

The key advantage of this approach is that instead of giving focus only to the splitting of a sentence into phrases, it has the intelligence to identify the actions that the original sentence includes and inherits it to the segments.

7

The order of precedence of the output segments is maintained based on the semantic structure of a sentence.

8

The primary focus of this segmentation approach on the Testing domain is due to the wide scope of implementing similar prepositional/conjunction-based NLP techniques in test case documents.

Conclusion

The solution discussed here depicts a method that leads to a huge reduction in the complexity present in preprocessing tasks of a test case document. With this, significant human effort in creating automation scripts or rules for accomplishing similar goals can be saved. The solution showed an innovative mechanism to segment a sentence on conjunctions and prepositions without breaking the order or meaning of segmented units. The preservation of relevant actions in all the output segments makes it a best fit for AI-based keyword or data-driven test automation frameworks. The solution is an independent module that can be plugged into any domain based on the segmentation requirements.

Reference

Software Testing Life Cycle <https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle>

Falcon Automation Framework <https://www.hcltech.com/white-papers/engineering/falcon-automation-framework>

Neural Network https://en.wikipedia.org/wiki/Neural_network

A Simple Introduction to Natural Language Processing <https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32>

A Comparative Analysis of Machine Comprehension Using Deep Learning Models in Code-Mixed Hindi Language https://link.springer.com/chapter/10.1007/978-3-030-12500-4_19

Sentence Segmentation <https://port.sas.ac.uk/mod/book/view.php?id=583&chapterid=383>

Deep Segment 2.0: Multilingual text segmentation with vector alignment <https://medium.com/@praneethbedapudi/deepsegment-2-0-multilingual-text-segmentation-with-vector-alignment-fd76ce62194f>

Transforming Test Cases to Action Driven Format Using NLP <https://www.hcltech.com/white-papers/engineering/transforming-test-cases-action-driven-format-using-nlp>

Part-of-speech tagging https://en.wikipedia.org/wiki/Part-of-speech_tagging

Author Info



Sujith Viswanathan

Sujith Viswanathan is a Senior Software Engineer with Post Graduate Degree in Data Science. He currently pursues a Post Graduate Diploma in Blockchain Technology and Management. He has been in the field of Artificial Intelligence for the past 3 years by providing solutions focused on Natural Language Processing, Computer Vision, and Speech Recognition using Deep Learning and Machine Learning techniques. Some of his research publications include:

1

“A Comparative Analysis of Machine Comprehension using Deep Learning Models in Code-mixed Hindi Language.”

2

“A Sequence based Machine Comprehension Modelling using LSTM and GRU.”

3

“Detection of Duplicates in Quora and Twitter Corpus.”



Srihari V

Srihari has been in Telecom and networking for past 16 years. He has managed various testing teams and has been creating next gen solutions as value adds for leading OEM clients. He is currently part of Solutions team and generate AI based solutions to support business needs.

ABOUT HCL TECHNOLOGIES

HCL Technologies (HCL) is a leading global IT services company that helps global enterprises re-imagine and transform their businesses through digital technology transformation. HCL operates out of 32 countries and has consolidated revenues of US\$ 6.97 billion, for 12 months ended 31st December, 2016. HCL focuses on providing an integrated portfolio of services underlined by its Mode 1-2-3 growth strategy. Mode 1 encompasses the core services in the areas of Applications, Infrastructure, BPO, and Engineering & R&D services, leveraging DRYICE™ Autonomics to transform clients' business and IT landscape, making them 'lean' and 'agile'. Mode 2 focuses on experience-centric and outcome-oriented, services such as Digital and Analytics Services (BEYONDigital™), IoT Works™, Cloud and Security, utilizing DRYICE™ Orchestration to drive business outcomes and enable enterprise digitalization. Mode 3 strategy is ecosystem-driven, creating innovative

IP-partnerships to build products and platforms business.

HCL leverages its global network of integrated co-innovation labs, and global delivery capabilities to provide holistic multi-service delivery in key industry verticals including Financial Services, Manufacturing, Telecommunications, Media, Publishing, Entertainment, Retail CPG, Life Sciences Healthcare, Oil & Gas, Energy & Utilities, Travel, Transportation & Logistics and Government. With 120,000 professionals from diverse nationalities, HCL focuses on creating real value for customers by taking 'Relationships Beyond the Contract'. For more information, please visit www.hcltech.com.

ABOUT HCL ENTERPRISE

HCL is a \$9.9 billion leading global technology and IT enterprise comprising two companies listed in India - HCL Technologies and HCL Infosystems. Founded in 1976, HCL is one of India's original IT garage start-ups. A pioneer of modern computing, HCL is a global transformational enterprise today. Its range of offerings includes product engineering, custom & package applications, BPO, IT infrastructure services, IT hardware, systems integration, and distribution of information and communications technology (ICT) products across a wide range of focused industry verticals. The HCL team consists of over 150,000+ ideapreneurs of diverse nationalities, who operate from 46 countries including over

500 points of presence in India. HCL has partnerships with several leading global 1000 firms, including leading IT and technology firms. For more information, please www.hcl.com



www.hcltech.com

Hello there! I am an Ideapreneur. I believe that sustainable business outcomes are driven by relationships nurtured through values like trust, transparency and flexibility. I respect the contract, but believe in going beyond through collaboration, applied innovation and new generation partnership models that put your interest above everything else. Right now 150,000+ Ideapreneurs are in a Relationship Beyond the Contract™ with 500 customers in 46 countries.
How can I help you?

HCL