TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 4329 - 4335

Multi-Document Abstractive Text Summarization through Semantic Similarity Matrix for Telugu Language

D Naga Sudha[1], Dr.Y Madhavee Latha[2]

[1]Research Scholar, JNTUH College of Engineering, Hyderabad, Telangana, India

[2]Prof,Malla Reddy Engineering College for Women, Telangana, India

Article Info
Volume 82
Page Number: 4329 - 4335
Publication Issue:
January-February 2020

Article History
Article Received: *18 May 2019*
Revised: *14 July 2019*
Accepted: *22 December 2019*
Publication: *21 January 2020*

Abstract

Telugu is one of the popular south Indian languages which is currently spoken by 84 million population in Telangana and Andhra Pradesh. Text summarization is an area of research with a goal to provide short text from huge text documents. Extractive text summarization methods have been extensively studied by many researchers. There are various type of multi document ranging from different formats to domains and topic specific. With the application of neural networks for text generation, interest for research in abstractive text summarization has increased significantly. This approach has been attempted for Telugu language in this article. Recurrent neural networks are a subtype of recursive neural networks which try to predict the next sequence based on the current state and considering the information from previous states. The use of neural networks allows generation of summaries for long text sentences as well. The work implements semantic based filtering using a similarity matrix while keeping all stop-words. The similarity is calculated using semantic concepts and Jiang Similarity and making use of a Recurrent Neural Network (RNN) with an attention mechanism to generate summary. ROUGE score is used for measuring the performance of the applied method on Telugu Language.

Keywords: *Abstractive Text Summarization, Multi-document, Text Generation, Semantic Role Labeling, Semantic Similarity Matrix, Semantic Selection, ROUGE, Summary Generation.*

I. INTRODUCTION

Text summarization is considered as an active field of research in natural language processing and can be classified into three main categories: extraction, compression and abstraction-based methods. Further summarization can be done on a single document with fixed or variable sentence length or size and also multiple documents which may be homogeneous or heterogeneous. Extraction based approaches involve source sentences and phrases to make the summary. These methods are easier to implement and provide valid summaries compared to the other two methods but require sentences to be selected efficiently and balancing between salience and redundancy. Compression based methods remove words or phrases to remove redundancy but

January - February 2020
ISSN: 0193 - 4120 Page No. 4329 - 4335

fail to merge sentences from different sources. Abstraction based summarization methods involve some form of natural language generation where the final summary consists of new words which are not present in the vocabulary of the source data. In recent years research in abstractive summarization has gained more attention due to the advancement in technology and it was found that the use of neural networks have improved performance and is preferred for automatic summarization.

Multi-document summarization can be based on either the document having different types of information in the document or having multiple documents with some common information. Multi-documents can also be of different types file formats. The documents can be further classified as homogeneous or heterogeneous. Homogeneous multi documents usually cover the same file type and context whereas heterogeneous cover multiple domains or topics. There are various homogenous multi documents available such as Document Understanding Conference (DUC) datasets, Text Analysis Conference (TAC) datasets and Minimum Data Set (MDS). Heterogeneous multi-document datasets are very few such as the heterogeneous multi-genre corpus called heterogenous MDS "hMDS" and subsequently auto-hMDS which are generated from summaries on Wikipedia and crowd-sourcing [1].

Semantic information approach is taken for abstractive summaries for more multi documents with semantic data, more specifically the graph based approaches. By first extracting sentences and using semantic role labeling to construct a semantic graph and then using a ranking algorithm to obtain key sentences and finally generating the summary [2-7]. Extractive and abstractive models can be combined with recursive neural networks (RNN) further for handling long text summarizations [8].

II LITERATURE SURVEY

While looking at the source documents we come across two types, homogeneous multi documents usually cover the same file type and context whereas heterogeneous cover multiple domains or topics. There are various homogenous multi documents available such as Document Understanding Conference (DUC) datasets, Text Analysis Conference (TAC) datasets and Minimum Data Set (MDS). Heterogeneous multi-document datasets are very few such as the heterogeneous multi-genre corpus called heterogenous MDS "hMDS" and subsequently auto-hMDS which are generated from summaries on Wikipedia and crowd-sourcing.

In terms of the different methods of text summarization, using neural networks have benefitted in extractive methods for handling semantics as well as redundancy compared to other traditional methods but lack in coherence compared to abstractive methods. Among abstractive summarization there are different approaches such as linguistic based approaches, semantic graph based approaches and hybrid extractive/abstractive approaches. Linguistic based approaches make use of syntactic representations and tree structures but lack abstraction to semantic meanings. Semantic graph based approaches focus on semantic role labeling to determine abstraction of input to core meaning to form graphs to filter out redundancy followed by text generator to build summaries as discussed in previous papers. Hybrid approaches make use of extractive methods to obtain an output summary to be fed to a text generator to build non-key words and phrases to further improve coherence and readability of sentences. One type of neural network is the recursive neural network which makes use of the same weights recursively on a structured input to predict the output. One subclass of the recursive neural network is the recurrent neural network which has a linear chain structure whereas a recursive model has a hierarchical structure. Recurrent neural networks also work on linear progression of time and make use of previous time steps and states during the current time step thus making this model lucrative for text generation. Use of recurrent neural networks with an attention

January - February 2020
ISSN: 0193 - 4120 Page No. 4329 - 4335

mechanism further helps with long text dependencies demonstrated by other researchers. Semantic information approach is taken for abstractive summaries for more multi documents with semantic data, more specifically the graph based approaches. By first extracting sentences and using semantic role labeling to construct a semantic graph and then using a ranking algorithm to obtain key sentences and finally generating the summary.

III IMPLEMENTATION DETAILS

Consider a homogeneous multi-document dataset D consisting of different documents of the same file type with sentences of different lengths. The documents will be on a particular context. A summary has to be generated of size significantly smaller to the overall size of the source documents consisting of generated words which maintain salience, coherence and accuracy using while avoid redundancies. Using ROUGE scores, precision and f-scores to measure the same. The complete detailed approach followed in this work is shown in below figure 1.

The documents are broken into sentences. Each sentence has the document number and its position in the document also attached to the sentence. Once all sentences have been extracted they are fed to next step for semantic role labeling. Semantic word phrases are identified which are also called as semantic arguments. Semantic arguments are grouped into two categories; core arguments consisting of subject, object and indirect object. Adjunctive arguments are location and time for predicate verb. All complete predicates related to the single sentence structure are considered to avoid loss of important terms contributing to salience and actual predicate of the sentence. Sentences containing more than one predicate are considered as composite predicate argument structures. After forming the predicate argument structures (PAS), they are split into tokens followed by the removal of stop words. Remaining tokens are stemmed and POS tagger is used to tag the terms of the semantic arguments. Tokens of noun, verb, location and time are extracted and move to next phase of similarity matrix construction. The similarities of the PAS are calculated pair wise based on noun-noun, verb-verb, location-location and time-time using Jiang's semantic similarity measure. By Jiang's similarity, the similarity of two concepts is dependent on the information shared by them. It calculates the semantic distance between any two concepts using below equation:

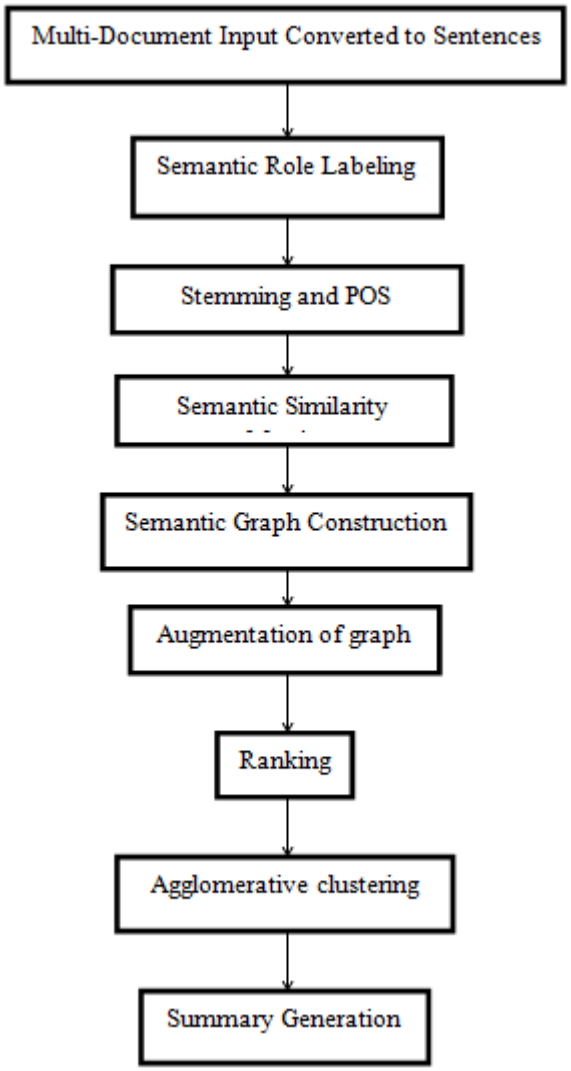

Figure 1: Block Diagram of the System Developed

$$Jiang(C1, C2)IC(C1) + IC(C2) - 2 * IC(Iso(C1, C2)) \quad (1)$$

Jiang's measure uses WordNet where Iso is the least common sub sum of the two concepts and then

January - February 2020
ISSN: 0193 - 4120 Page No. 4329 - 4335

determines the information content(IC) of the concepts through probability of occurrence in the corpus as below,

$$IC(C) = -\log P(C) \quad (2)$$

$$P(C) = \frac{Freq\,(C)}{N} \quad (3)$$

Where P(C) is the probability and Freq(C) is the frequency of concept 'C', N stands for maximum number of nouns. The semantic similarity matrix is built from the similarity scores calculated from each pair.

The next step is constructing an undirected weighted graph from the similarity matrix where self-transition are given weight 0 to avoid them and others are given weight greater than 0. A link is established above a similarity threshold of 0.5.

Next the semantic similarity graph is augmented with document relationship based on PAS semantic similarity to the title, position and document. Genetic algorithms can be used to obtain optimal feature weights. The ranking step performs ranking based on edge weights of the graph along with salience score. After ranking is performed, agglomerative clustering is done to remove redundancy.

The source sequence $S = [s_1, s_2 \ldots, s_m]$ is converted into fixed length vector c by the encoder. If h_t is the state at time t, c is the context vector then f is a dynamic function and $\emptyset$ summarizes the hidden states,

$$h_t = f(s_t, h_{t-1}) \quad (4)$$

$$c = \emptyset(\{h_1, h_2, \ldots, h_m\}) \quad (5)$$

BRNN processes input sentences in two hidden layers, one for forward and one for backward [8]. For each position, the forward and backward hidden states are concatenated into final hidden state. For each position 'i', $h_i^{\rightarrow}$is forward hidden state and $h_i^{\leftarrow}$ is backward hidden state then final hidden state h_i is,

$$h_i = h_i^{\rightarrow} \oplus h_i^{\leftarrow} \quad (6)$$

The decoder unfolds c into the target sequence.

$$d_t = f(y_{t-1}, d_{t-1}, c)\; p(y_t|Y_{<t}, S) = g(y_{t-1}, d_t, c) \quad (7)$$

Where $d_t$the RNN state at time t is, y_t is predicted target word at time t through function g, $y_{<t}$ denotes the history. The decoder classifies the vocabularies in order to optimize the loss function. After the vector passes through the softmax function the word with highest probability will be the output.

The attention function is used to reduce the load to summarize entire source into a fixed length vector as context. It uses dynamically changing context c_t at the time of generating t^{th} target word,

$$c_t = \sum_{i=1}^{m} \alpha_{ti} h_i \quad (8)$$

$$\alpha_{ti} = softmax(Z_\alpha^T \tanh(W_\alpha d_{t-1} + U_\alpha h_i)) \quad (9)$$

Where h_i is the hidden state of the encoder, α_{ti} gives how much i^{th} word from the original sequence contributes to generating the t^{th} word in the summary, Z_α^T is the weighting vector, W_α and U_α are weighting matrices.

The pointer mechanism is used to prevent buffer overflow. It is applied in a switch manner between generator which generates new words and pointer which extracts the word from the source. The switch is trained in sigmoid activation function over linear layer. The probability of switch turning pointer is,

$$P(d_i) = \sigma(Z^T(W_h h_i + W_e E[o_{i-1}] + W_c c_i + b)) \quad (10)$$

Where $E[o_{i-1}]$ is the embedding vector of the emission of the previous time, c_i is the attention-weighed context vector, h_i is the hidden i^{th} time state, W_h,W_e,W_c, b and Z are model parameters.

The dataset can be obtained from various sources such as DUC, TAC, the International Computer Science Institute (ICSI) corpus and more mentioned in [9] or can be built like [1] from Wikipedia or other various websites.

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 4329 - 4335

Documents can hold data in various formats like audio, text and visual images. When the input to the NLP problem consists of multiple such documents we use the term multi-document. If the text formats, file types, context or topics vary from the various documents then the documents are considered as heterogeneous. However, if the documents have the same format, type, context or topics then they are considered as homogeneous. A single document can also be considered as multi document based on the internal contents of the document. Documents can also be restricted to having the same size or different size in terms of length or file size. Heterogeneous multi documents may also include multiple languages in which case become multi-lingual heterogeneous multi documents which would require some form of require machine translation.

Application of text summarization can be seen on search engines and along the domains of academic, business, news journalism and medicine where the user requires a brief readable gist of a large amount of documents in order to understand the information contained in the corpus of data. Search engines make use of query based text summarization to go through vast number of documents on the internet in order to provide a small gist along with the links relevant to the query. An example would be Google providing a brief gist based on search query from the document with the link. In academia, summaries of large amount on documents such as various books and papers in a coherent and readable text to save time of going through every document individually. News agencies use text summarization to form abstracts and headlines worthy of being short while having the important information within a certain number of words for their various pages or websites. Businesses use text summarization in order to obtain required information in time to make decisions. In the domain of medicine text summarization may be used in determining illnesses and symptoms by doctors along with required medication.

IV OBTAINED RESULTS

S1: కోచ్‌ను ప్రకటించేసిన బీసీసీఐ .., ఇంటర్వ్యూ ఎలా జరిగిందంటే..._ముంబై: అనేక వివాదాలకు ముగింపు పలకాలనే క్రమంలో భారత మహిళా క్రికెట్ కోచ్‌గా ఎవర్ని నియమించాలోనన్న క్రమంలో బీసీసీఐ సందిగ్ధానికి తెరదించింది. రామన్‌ను భారత జట్టు కోచ్‌గా నియమిస్తూ బీసీసీఐ అధికారికంగా ప్రకటించేసింది. కపిల్ దేవ్‌తో పాటుగా మరికొందరి నేత్యత్వంలో కమిటీ ఈ అభిప్రాయాన్ని బీసీసీఐకు సూచించింది. భారత మహిళల క్రికెట్ జట్టు కోచ్ ఎంపిక కోసం బీసీసీఐ గురువారం నిర్వహించిన ఇంటర్వ్యూలలో.. కోచ్ పదవికి మొత్తం 28 మంది కోచ్ పదవికి దరఖాస్తు చేసుకోగా.. బీసీసీఐ పది మందిని ఇంటర్వ్యూలకు ఆహ్వానించింది.

S2: అగ్ర స్థానాన్ని పదిలం చేసుకున్న కోహ్లీ, టాప్ 10లోకి నాథన్_న్యూడిల్లీ: ఐసీసీ తాజాగా విడుదల చేసిన ర్యాంకింగ్స్‌లో టీమిండియా కెప్టెన్ విరాట్ కోహ్లీ అగ్రస్థానాన్ని పదిలం చేసుకున్నాడు. ఆస్ట్రేలియాతో జరిగిన రెండో టెస్టులో సెంచరీకి మించిన స్కోరు సాధించడంతో విరాట్ కోహ్లీ టెస్టు ర్యాంకింగ్స్‌లో అగ్రస్థానాన్ని సుస్థిరం చేసుకున్నాడు. అయితే రెండో స్థానానికి కోహ్లీకి కేవలం 19పాయింట్లు మాత్రమే తేడా ఉంది. పెర్త్ టెస్ట్ తొలి ఇన్నింగ్స్‌లో 123 పరుగులు చేసిన కోహ్లీ.. 934 పాయింట్లతో ఐసీసీ టెస్ట్ ర్యాంకింగ్స్‌లో తొలి స్థానంలో నిలిచాడు.

These sentences are then read and subjected to semantic role labeling after which they are added to a dictionary where the key is the sentence number. Semantic role labeling (SRL) is performed using the python implementation of SENNA done by practNLPtool which takes as input each sentence and outputs dictionaries with keys as various semantic roles and values as words. SRL tags are based on core and adjunctive arguments. The core arguments are classified as verb (V), subject (A0), object (A1), indirect object (A2), start (A3), end (A4) and direction (A5). Adjunctive arguments are direction (AM-DIR), manner (AM-MNR), location (AM-LOC), temporal (AMP-TMP), purpose (AM-PRP), negation (AM-NEG), reciprocal (AM-REC) and discourse (AM-DIS). The tool provides a list of all predicate argument structures associated with the sentence. A simple predicate argument structure consists of a single predicate along with at-least two or more arguments to form a sentence. There are also composite sentences consisting of many sub predicate argument structures along with the overall predicate argument structure. Here, for such sentences we consider only the overall predicate argument structure in such cases.

{'2': [{'A1': [[('కోచ్‌ను', 'DT')], [('ప్రకటించేసిన', 'NN')]], 'V': [[('బీసీసీఐ', 'VBG')]], 'A3': [[('ఇంటర్వ్యూ ఎలా', 'IN')], [('జరిగిందంటే', 'DT')], [('ముంబై', 'SYM')], [('అనేక', 'IN')], [('వివాదాలకు', 'VBN')], [('ముగింపు', 'NN')], [('పలకాలనే', 'IN')], [('28', 'SYM')], [('క్రమంలో', 'NN')], [('భారత మహిళా', 'VBG')], [('92', 'SYM')], [('క్రికెట్ కోచ్‌గా', 'NN')]]}, and so on.

'3': [{'A1': [[('అగ్ర స్థానాన్ని', 'EX')], [('పదిలం', 'VBZ')], [('చేసుకున్న', 'DT')], [('కోహ్లీ', 'SYM')], [('10, 'SYM')], [('లోకి', 'IN')], [('టాప్', 'NN')]], 'A0': [[('నాథన్_న్యూడిల్లీ:', 'JJ')], [('ఐసీసీ', 'NN')], [('తాజాగా', 'NN')], [('విడుదల', 'NNP')], [('చేసిన', 'NNP')]], 'V': [[('ర్యాంకింగ్స్‌లో', 'VBD')]], 'AM-LOC': [[('టీమిండియా', 'IN')], [('కెప్టెన్ విరాట్ కోహ్లీ', 'DT')], [(' ', 'NN')], [('అగ్రస్థానాన్ని', 'JJ')]], 'AM-TMP': [[('పదిలం చేసుకున్నాడు', 'RB')], [('ఆస్ట్రేలియాతో', 'IN')], [('జరిగిన', 'NN')], [('రెండో టెస్టులో', 'NNP')]]}} and so on.

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 4329 - 4335

The semantic similarity matrix is a sentence to sentence square matrix determined by the number of sentences in the input data. Further, each sentence to sentence similarity score is calculated based on the word-word semantic similarities in the sentence pair determined by the predicate argument structures.

In order to calculate the similarity of two sentences, the predicate argument structures of the two sentences are considered. A word similarity matrix is formed where if m is the number of unique words in sentence1 and n is the number of unique words in sentence2 then the word similarity matrix would be m×n shown in Table 1.

It has been shown that the jc-similarity is the closest to human similarity calculation taking into consideration the semantic concepts of words. Here each word can be considered as a concept and the Jiang-Conrath similarity is calculated as below equation

$$Jiang(C1, C2) = IC(C1) + IC(C2) - 2 * IC(Iso(C1, C2))$$

Here, keywords are the concepts C1, C2 being compared and the least common subsume lso is considered as the closest immediate parent of the two considered concepts which subsumes or contains them. The information content IC of concepts is calculated based on the probability of occurrence in a corpus by below equations,

Table 1: Semantic Similarity Matrix for the Individual Words

Similarity	అగ్ర స్థానాన్ని	పదిలం	చేసుకున్న	కోహ్లీ
కోచ్‌ను	0	0	0	0
ప్రకటించేసిన	0.0540739328	0.0765763519	0.0579670896	0.0577286886
బీసీసీఐ	0.0524244582	0.0944489407	0.0822580744	0.0661887831
ఇంటర్వ్యూ ఎలా	0	0	0	0
జరిగిందంటే	0	0	0	0
ముంబై:	0.0470718493	0	0.0581188824	0.0578792337
అనేక	0	0	0	0
వివాదాలకు	0	0.0962139702	0	0.0721244228
ముగింపు	0.0619305149	0.0856160121	0.0670911399	0.0667719902
పలకాలనే	0	0	0	0
క్రమంలో	0	0	0	0
భారత	0.0445472149	0	0.0543180550	0.0541086698
మహిళా	0	0	0	0
క్రికెట్	0	0	0	0
కోచ్‌గా	0	0	0	0

$$IC(C) = -\log P(C)$$

$$P(C) = \frac{Freq\,(C)}{N}$$

Where P(C) is the probability and Freq(C) is the frequency of concept 'C', N stands for maximum number of nouns.

V CONCLUSION

The below tables shows that with 60 epochs and 100 epochs we achieve reduction in summary From ROUGE results, we see an increase in F score, Precision and Recall scores for ROUGE-1 and ROUGE-2 and decrease in ROUGE-n. Overall we see significant reduction in text as well as formation of valid summary with further scope for optimization in terms of further augmenting the sentences with context data and augmenting weights with respect to entities and title information for Telugu documents. We considered Sports as the domain or context of text as the input for the system. Improvements can be made with respect to clustering method and ranking for sentence selection as well as training on larger document sets. Future scope would also involve using the model with different types of multi-document datasets. ROUGE score tend to require a reference summary for comparison and try to find the number of matches between the two summaries. This can cause low results in score if text is generated. Hence there is

January - February 2020
ISSN: 0193 - 4120 Page No. 4329 - 4335

scope for building automated metric to calculate based on semantic matches from output and reference summaries for other regional languages of India.

Table 2: Obtained Results for the Sports Documents

Sample	Context	No. of documents	No. of line	No. of words	No. of epochs	summary lines	summary words
1	Sports1	3	86	1961	50	20	218
					1000	18	229
2	Sports2	5	145	2768	50	42	731
					1000	37	675

Table 3: ROUGE Results

Sample	Context	ROUGE1			ROUGE2		
		F	P	R	F	P	R
1	Sports1	0.1630	0.1342	0.2349	0.0234	0.0222	0.0233
		0.17631	0.1095	0.2340	0.0332	0.0267	0.0453
2	Sports2	0.16432	0.12150	0.3108	0.0378	0.0145	0.0739
		0.18210	0.1324	0.3561	0.0432	0.0230	0.1345

REFERENCES

[1] Benikova, D., Mieskes, M., Meyer, M. C., & Gurevych, I. (2016). Bridging the gap between extractive and abstractive summaries:Creation and evaluation of coherent extracts from heterogeneous sources. the 26th International Conference on Computational Linguistics. Osaka, Japan.

[2] Dernoncourt, F., Ghassemi, M., & Chang, W. (2018). A Repository of Corpora for Summarization. LREC 2018, Eleventh International Conference on Language Resources and Evaluation. Miyazaki, Japan.

[3] Fang, Y., Zhu, H., Muszyńska, E., Kuhnle, A., & Teufel, S. (2016). A Proposition-Based Abstractive Summariser. The 26th International Conference on Computational Linguistics. Osaka, Japan.

[4] Kallimani, D. S., & S, R. N. (2017). Abstractive Multi-Document Summarization. IEEE.

[5] Khan, A., Salim, N., & Farman, H. (2016). Clustered Genetic Semantic Graph Approach for Multi-Document Abstractive Summarization. IEEE.

[6] Li, W., He, L., & Zhuge, H. (2016). Abstractive News Summarization based on Event Semantic Link Network. The 26th International Conference on Computational Linguistics. Osaka, Japan.

[7] S, A., John, A., & Nath, A. G. (2017). Multi-document Abstractive Summarization Based on Predicate Argument Structure. IEEE.

[8] Wang, S., Zhao, X., Li, B., Ge, B., & Tang, D. (2017). Integrating Extractive and Abstractive Models for Long Text Summarization. IEEE 6th International Congress on Big Data.

[9] Zopf, M., Peyrard, M., & Eckle-Kohler, J. (2016). The Next Step for Multi-Document Summarization: A Heterogeneous Multi-Genre Corpus Built with a Novel Construction Approach. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics. Osaka, Japan.