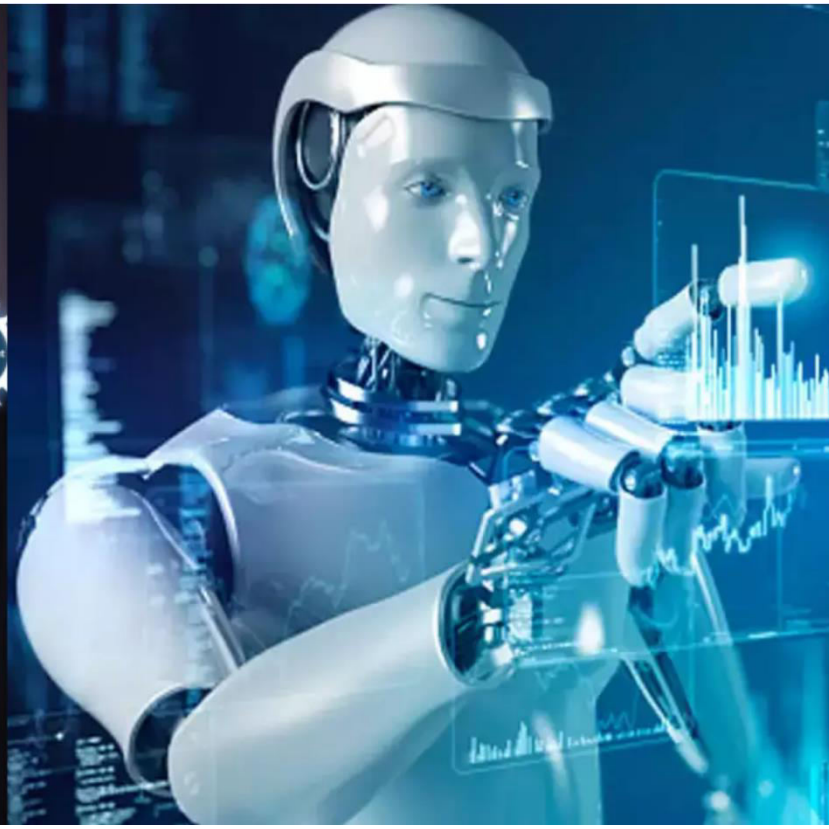
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 9, September 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

Sentence Compression using Natural Language Processing

Ravi Malikdin Saroj, Rajesh D Wagh

Department of Computer Science and Engineering, SYCET, Aurangabad, India

ABSTRACT: Using extractive text summarization, we attempt sentence compression in this study. As a result, we can train a deep learning model to complete the job by rewriting it as a multi-label deletion-based issue. We can create summaries that are mainly grammatically sound and useful with fewer training examples and yet earn F1 scores that are comparable to those from previous exams. Similar outcomes were obtained using a (Proof of Concept) POC of the model on a confidential dataset held by the author's employer. We describe a unique unsupervised sentence compression approach that uses Stanford Typed Dependencies to extract information items and an NLP sentence compression engine to produce compressed phrases. An automated examination demonstrates that our strategy yields superior outcomes.

KEYWORDS: Natural Language Processing, Sentence Compression, Text Summarization

I. INTRODUCTION

It is becoming more and more crucial to find effective techniques to analyze and get insights from textual data in this era where its volume is expanding exponentially. Automatic text summarizing is required to decrease the total processing time for textual data since it is time-consuming and challenging for humans to extract essential points from enormous quantities of text. Text summarizing seeks to condense lengthy texts into a concise, fluid summary that solely emphasizes the text's important ideas. There are typically two methods for solving the issue:

1) Using important words or phrases from the source text and combining them to create a summary is known as extractive text summarizing.
2) To create an abstractive text summary, portions of the original text are paraphrased and condensed. As it is difficult to programmatically analyze an abstractive summary because there is no restriction on the language and phrase types that can be employed, extractive text summarizing is used in this study instead of abstractive text summarization.

The TextRank algorithm, which extracts phrases most representative of the text based on similarity scores across sentences, and constructing a summary with terms with the greatest TFIDF scores are examples of previous techniques for extractive text summarizing. Instead, we use a different strategy and train a textual summarizer that is based on sentence compression using deep learning methods. To maintain the grammatical coherence and significant substance of the original text, this takes the form of a multi-label deletion-based task where the model chooses which words to keep from the original text. Thus, summaries produced using our approach are logically a subset of the original text.

The task of sentence compression falls within the category of text-to-text creation and is defined as follows: What subset of the words in a phrase S that contains the letters w_1, w_2...w_n is grammatically correct and retains the meaning of S? A powerful compression system would be useful for various applications, including news digests, compression for mobile devices with small screens, and subtitle production. Since the majority of text and speech summarization algorithms are currently extractive, phrase compression techniques are frequently used to reduce output redundancy.

Many methods for condensing sentences have been developed in recent years [11] (Jing, 2001; Knight & Marcu, 2022; Gagnon & Da Sylva, 2015; Turner & Charniak, 2015; Clarke & Lapata, 2023).

To generate grammatical output, many explicitly rely on a language model, often a trigram model (Knight & Marcu, 2022; Hori & Furui, 2022; Turner & Charniak, 2015; Galley & McKeown, 2017).

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

Working with a language like English, which has rigorous word order, justifies testing the output's grammaticality using a language model, and all but one of the approaches above have been used with English data. However, a more thorough study to assess grammaticality is required when condensing sentences in languages with less rigorous word order. Furthermore, the trigram model disregards sentence structure, which has the potential to drastically alter the meaning of the source phrase even for languages with strict word order.

A thorough vocabulary is needed (Jing, 2001) or manually created rules (Gagnon & Da Sylva, 2015; Clarke & Lapata, 2023) to identify problematic elements in approaches that go beyond the word level. A vocabulary is not always accessible, and hand-crafted rules may not be comprehensive enough to apply in every situation (e.g., PPs can be trimmed).

In this research, we describe a unique unsupervised method of sentence compression that is inspired by the idea that compressing trees can more effectively guarantee the output's grammaticality. In particular, given a dependency tree, we wish to remove any subtrees that neither add significant context to the sentence's content nor are required syntactic arguments. A tree pruning technique is unlikely to yield compression with a completely different meaning and does not create new dependencies. Our method is unsupervised and flexible enough to work with other languages as long as a dependency parser and corpus are provided for those languages.

II. RELATED WORK

It is becoming more and more crucial to find effective techniques to analyze and get insights from textual data in this era where its volume is expanding exponentially. Automatic text summarizing is required to decrease the total processing time for textual data since it is time-consuming and challenging for humans to extract essential points from enormous quantities of text. Text summarizing seeks to condense lengthy texts into a concise, fluid summary that solely emphasizes the text's important ideas. There are typically two methods for solving the issue:

a) Using important words or phrases from the source text and combining them to create a summary is known as extractive text summarizing.
b) To create an abstractive text summary, portions of the original text are paraphrased and condensed. As it is difficult to programmatically analyze an abstractive summary because there is no restriction on the language and phrase types that can be employed, extractive text summarizing is used in this study instead of abstractive text summarization.

Creating a summary using terms with the greatest TFIDF scores and using the TextRank algorithm, which extracts the sentences that are the most representative of the text based on similarity scores between sentences, are examples of earlier techniques for extractive text summarizing. Instead, we use a different strategy and employ deep learning to train a textual summarizer that is based on sentence compression. As a result, it takes the shape of a multi-label deletion-based job where the model chooses which words from the original text to retain while attempting to maintain the grammatical consistency and significant substance of the original text. As a result, summaries produced using our technology are inextricably linked to the original text.

Constrained compression is not well suited to current methods. Although approaches based on integer linear programming (ILP) can easily accommodate such lexical and length restrictions (Clarke and Lapata, 2023; Filippova and Altun, 2013; Wang et al., 2017), these methods rely on sluggish third-party solvers to optimise an NPhad integer linear programming objective, which adds to user wait time.

A different LSTM tagging method (Filippova et al., 2015) prohibits practitioners from specifying length or lexical limitations and necessitates the purchase of a pricey graphics processing unit (GPU) to achieve minimal runtime latency (access to GPUs is a hurdle in disciplines like social science and journalism).

These shortcomings prohibit the implementation of present compression approaches in search user interfaces, where length, lexical, and latency requirements are crucial (Marchionini, 2016; Hearst, 2021). Thus, we provide a novel stateful query-focused compression technique.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

Knight and Marcu (2000), Clarke and Lapata (2023), Filippova et al. (2015), and Wang et al. (2017) are a few examples. To our understanding, this study is the first to take strict length and lexical limitations into account when thinking about extractive compression.

We contrast the VERTEX ADDITION method with ILP-based compression techniques that condense phrases using an integer linear programming objective (Clarke and Lapata, 2023; Filippova and Altun, 2013; Wang et al., 2017). ILP techniques need worst-case exponential computation, but they can readily handle lexical and financial limits with extra optimization requirements.

Finally, LSTM tagger-based compression techniques (Filippova et al., 2015) are unable to enforce lexical or length constraints at this time Future research may employ and adapt limited-generation approaches to overcome this issue.

(Gehrmann et al., 2018; Post and Vilar, 2018; Kikuchi et al., 2016). 3 Using VERTEX ADDITION to compress With comparable methods in transition-based parsing as our model (Nivre, 2003), we describe a new transition-based technique for condensing sentences under lexical and length restrictions. Because it grows a (potentially unconnected) subgraph in the dependency parsing of a phrase, one vertex at a time, our method is known as VERTEX ADDITION. This method may create restricted compressions using a linear algorithm, resulting in latency that is 11 times lower than that of ILP approaches (x4). To the best of our knowledge, our technique is also the first to create 1Some approaches (Rush et al., 2015; Mallinson et al., 2018) compress through creation rather than deletion.

According to Chuang et al. (2012), the extractive technique is designed for reliable, usable, and practical search systems. Users might not trust abstractive summaries, especially when there is a semantic inaccuracy (Zhang and Cranshaw, 2018).

When choosing tokens, ILPs are exponential in jV j (Clarke and Lapata, 2023) and exponential in jEj (Filippova et al., 2015). compressions in a parse by including vertices as opposed to removing them (Knight and Marcu, 2000; Almeida and Martins, 2013; Filippova and Alfonseca, 2015). The author makes the boolean relevance model assumption that S must include Q and reserves more complex relevance models for future research.

To check the output's grammaticality, several current compression algorithms employ a noisy-channel method (Knight & Marcu, 2022; Turner & Charniak, 2015; Galley & McKeown, 2017). Utilising a subcategorization lexicon (Jing, 2001) or defining a list of usually probable elements are other techniques to check grammaticality and determine whether a constituent is required or may be trimmed.

Gagnon and Da Sylva (2015) trim dependency trees by eliminating noun appositions, subordinate clauses, and verbal prepositional complements. This does not always ensure grammaticality. Additionally, it could remove crucial data from the tree. The majority of methods are supervised and call for training data to determine which parts of a phrase may be eliminated (Riezler et al., 2003; McDonald, 2016). Nevertheless, obtaining training data is challenging. There aren't many unsupervised ways, though.

For instance, Hori & Furui (2022) offer a scoring system that depends on data from the language model and word significance score. Dynamic programming is then used to find a compression that maximizes the scoring function for a given length.

Another unsupervised strategy is presented in Clarke & Lapata's (2023) work. The job is formulated as an optimization problem, and integer linear programming is used to resolve it. Their goal function is influenced by two scores: a word importance score and a trigram language model score. Additionally, several linguistic restrictions, such as those dictating which arguments should be maintained, are used to guarantee the output's grammaticality. In this work, we propose a technique that compresses dependency trees rather than strings of words as an alternative to the common language model base for compression systems. We shall contend that the benefits of our formulation are as follows: First off, the method is unsupervised; the only prerequisites are a suitably sizable corpus and a dependency parser. Second, determining which arguments are required doesn't need a hand-crafted set of rules or a sub-categorization vocabulary. Thirdly, by considering word significance and syntax, it determines a compression that is globally optimum.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

III. PROPOSED SYSTEM

In this research, we describe a unique unsupervised method of sentence compression that is inspired by the idea that compressing trees can more effectively guarantee the output's grammaticality. Specifically, given a dependency tree, we wish to remove any sub-trees that neither provide an essential syntactic argument nor add to the sentence's content. A tree pruning technique is unlikely to yield compression with a completely different meaning and does not create new dependencies. Our method is unsupervised and flexible enough to work with other languages as long as a dependency parser and corpus are provided for those languages. On English data sets, we will test our methodology and get results that are on par with or better than the state of the art.

By removing dependence edges from a sentence's dependency tree, our approach reduces sentences. The goal is to maintain dependencies that are either necessary for the output to be grammatically correct or have a significant term as the dependant. Tree transformation, tree compression, and tree linearization are the first three stages of the method.

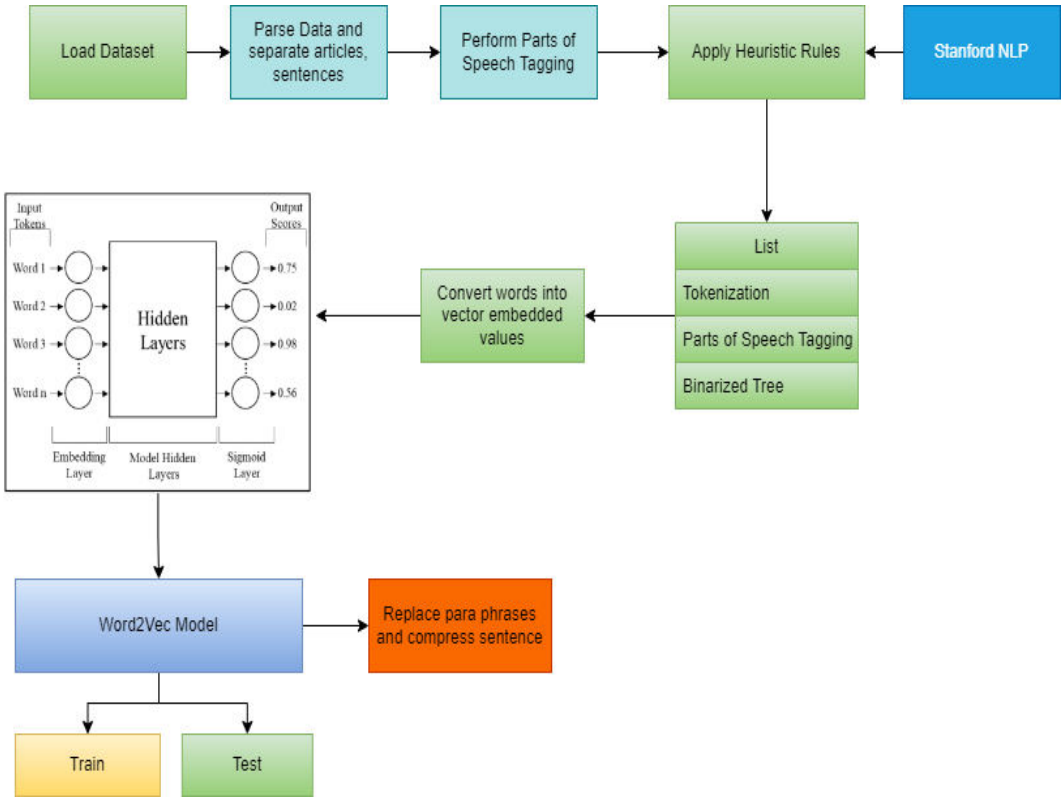


Fig.1. Graphical Abstract of the Proposed System Architecture

A dependency tree is altered before compression, or before some dependencies are eliminated. With the following phrase, we shall illustrate how the transformations work:

(1) He said that he lived in Paris and Berlin

One explicit root node is inserted by the initial transformation (ROOT). The second transformation (VERB) has the effect of giving each inflected verb in the tree an edge coming from the root node. The s label is present on every edge that leaves the root node.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

In addition, we eliminate auxiliary edges and memories of grammatical details like verb voice, tense, and negation. The remaining modifications aim to clarify the relationships between open-class terms. We want to choose whether to prune an edge based on two factors:

i. how relevant the argument is for the head; and
ii. how informative the dependent term is. Consider the source sentence from (2) as an illustration.

(2) After some time, he moved to London.

Checking whether an argument associated with the label pp is required for the verb move would not be very useful. A closer examination of the preposition "after" as opposed to "to" might be more instructive. Prepositional nodes are removed and placed as labels on the edge from their head to the appropriate noun in the PREP transformation (Fig. 1(c)) in response to this. A chain of conjoined components is also broken down, and each piece is attached to the first element in the chain's head with the label of the initial element.

In this manner, if just one of the conjoined parts is of interest to us, we don't have to maintain the entire sequence of them during compression. Verbs are not subject to this most recent transformation.
Resulting statement:

He lived in Paris and Berlin and after some time moved to London

Integer linear programming is used to tackle the optimization issue that we express as the compression job. We choose which dependence edges to eliminate from a changed dependency tree (or a graph if additional edges have been introduced).

According to intuition, we cannot eliminate necessary dependencies from the tree because of the conditional probability. For instance, because P(subj|work) is greater than P(with work), the subject will be kept intact but the prepositional label and the entire PP can be removed. By doing this, we may avoid having to add a constraint for each mandatory parameter (such as a subject or direct object). We don't also need a sub-categorization lexicon to search up the mandatory arguments for a given verb. Because the dependence edges, with which verb arguments are tied to the head, receive high marks, verb arguments are kept.

Noting that we could merely compute prepositions, which would not be very helpful if we did not perform the PREP transformation, we would not be able to discriminate between distinct prepositions. Since we calculated the conditional probability given a word lemma, the probabilities for English are lower than those for English. The collection of potential labels for a node is, on average, bigger than in English since the part of speech information cannot be inferred from the lemma in English.

The compressed sub-tree is always rooted in the node that was added as a result of the initial transformation because of the restriction. If the structure above the embedded clause is not preserved, compressing a sentence to an embedded clause is not feasible. But more often than not, an embedded clause is more significant than the primary sentence. For instance, in the line He stated they must be kept in Beirut, the embedded clause is the part to which the source sentence should be compressed since it contains the information. This issue is exactly what the VERB change is intended to fix. We may condense the source phrase to any clause since every inflected verb has an edge from the root node.

Presenting the words of a compressed sentence in the original sentence's sequence is a very basic but effective linearization approach. We linearize the trees created for the English data using this approach, which has been used before. Unfortunately, due to English's restrictions on word order, this strategy cannot be directly applied to the language. According to a rule of English grammar, the inflected component of the verb is to be in the second place in the main phrase, with precisely one constituent in the first position. As a result, the verb becomes the first part of the sentence when a constituent that was pruned off due to compression occupies the sentence start position in the source phrase, producing an unacceptable output. There are English-specific linearization techniques that can determine the best wording for a statement.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

About Dataset

English Compression Corpus:

The English data we utilize comes from a document-based compression corpus that includes 82 news articles from the British National Corpus and the American News Text Corpus. The Stanford PCFG parser (Klein & Manning, 2003) and RASP (Briscoe et al., 2016) will be used to parse the corpus. The latter offers the option to transform the result into a dependency format, which we employ (de Marneffe et al., 2016), whereas the former produces a collection of dependence relations as its output.

In addition to the corpora mentioned above, we also employ the Tipster corpus to determine word significance scores and conditional probabilities of syntactic labels given head lemmas. The 128 million words and verbs are used to determine the significance score. For conditional probabilities, a slightly smaller sample of Tipster (about 6 million tokens) is used in the computation. Comparable to the size of the data set we use to calculate the probability for English is the latter figure. There, we compute conditional probability and significance scores using a dataset of around 4,000 English Wikipedia articles.

The CDG parser (Foth & Menzel, 2016) was used to parse the corpus, which also has the same dependency structure as TuBa-D/Z (Versley, 2015). Although dependency relations are identified in both corpora, the annotation of the English and English data sets differs significantly. The Stanford parser converts the constituent's phrase into a dependency structure, giving the semantic head of the constituent syntactic head status. For instance, the root of the tree in the phrase He is right is the word right. The English corpora, however, always have a verb as the root of the phrase. We create a set of rules to make the verb the root of the tree in all situations to harmonies the forms.

IV. RESULTS AND DISCUSSION

Using the widely-used key index parameters (KIP) measures, we assessed performance. These measures have been demonstrated to be highly linked with human evaluations and are based on unigram, bigram, and unigram + skip-bigram overlap with a maximum skip distance of 4. KIP ratings can be used as a summary readability indicator. Although KIP scores do not require consecutive matches, word order is still important.

Our framework was created with the meeting industry in mind. In fact, redundancy is necessary for our creative component, the multi-sentence compression graph (MSCG), to function well. Meeting speech frequently has this redundancy, whereas traditional texts do not.

The MSCG is also noise-resistant by design, and our unique path re-ranking method based on graph degeneracy makes it even noise-resistant. Our framework benefits from ASR input as a result. Last but not least, we employ a language model to prioritise fluent pathways, which is critical when dealing with (meeting) speech but less so when handling well-formed data

Our framework creates a single, fully unsupervised end-to-end summarization framework and introduces some novel elements while combining the advantages of nlp and deep learning approaches that had previously been applied to three different tasks (keyword extraction, multi-sentence compression, and summarization). We produce sufficiently grammatical abstractive summaries while taking noisy utterances as input and not depending on any annotation or training data, according to a thorough examination on the AMI and ICSI corpora. Finally, our method is very immediately adaptable to languages other than English because it is completely unsupervised

Table 4.0 Comparison with existing system

Method	Accuracy	Recall	F Measure
No Compression	88.78 %	87.89 %	NA
Manual Compression	88.04 %	85.32 %	NA
AutoCompression	87.95 %	86.23 %	NA
TextRank	39.55 %	39.22 %	NA
Longest Greedy	37.31 %	38.56 %	NA
Our System	92.05 %	93.69 %	93.36 %

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

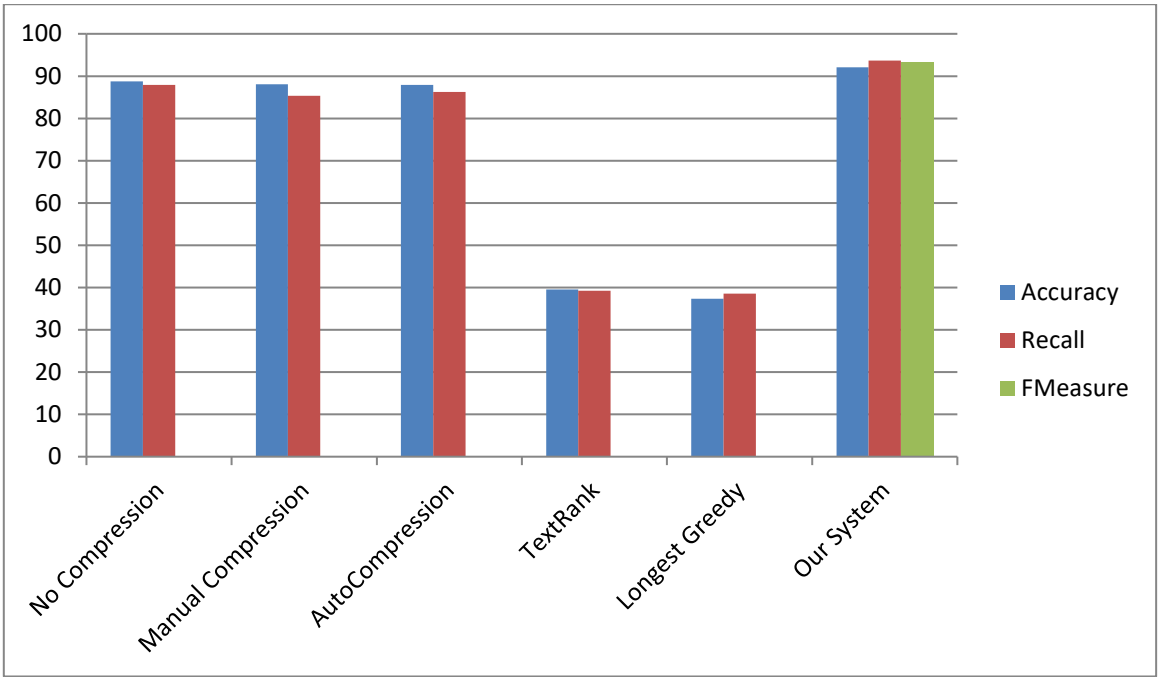


Fig 2.0 Graphical Representation and Comparison of various systems

For aspect-based sentiment analysis, the notion of employing sentiment sentence compression can be seen as a fundamental framework. We think that the emotion sentence compression model will be useful for more sentiment analysis applications that strongly rely on syntactic features.

V. CONCLUSION

We introduced a brand-new compression technique that tests grammaticality without using a language model and compresses dependency trees. The process is unsupervised and easily translatable to different languages. It determines which arguments may be pruned without the use of a sub-categorization lexicon or complex hand-crafted algorithms, and it finds a globally optimum compression that considers syntax and word significance. We showed how the parser affects the system's performance and offered a method to gauge how well-suited a parser is to the compression task. The outcomes suggest that the dependency-based strategy is a good alternative to the language model-based one.

REFERENCES

1. Briscoe, Edward, John Carroll & Rebecca Watson (2016). The second release of the RASP system. In Proceedings of the COLING-ACL Interactive Presentation Session, Sydney, Australia,2016, pp. 77–80.
2. Clarke, James & Mirella Lapata (2016). Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, 17–21 July 2016, pp. 377–385.
3. Clarke, James & Mirella Lapata (2023). Global inference for sentence compression: An integer linear programming approach. Journal of Artificial Intelligence Research, 31:399–429.
4. de Marneffe, Marie-Catherine, Bill MacCartney & Christopher D. Manning (2016). Generating typed dependency parses from phrase structure parses. In Proceedings of the 5th International Conference on Language Resources and Evaluation, Genoa, Italy, 22–28 May 2016, pp. 449–454.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409069|

5. Filippova, Katja & Michael Strube (2017). Generating constituent order in English clauses. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, 23–30 June 2017, pp. 320–327.
6. Foth, Kilian & Wolfgang Menzel (2016). Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, 17–21 July 2016, pp. 321–327.
7. Gagnon, Michel & Lyne Da Sylva (2015). Text summarization by sentence extraction and syntactic pruning. In Proceedings of Computational Linguistics in the North East, Gatineau, Quebec, ´ Canada, 26 August 2015.
8. Galley, Michel & Kathleen R. McKeown (2017). Lexicalized Markov grammar for sentence compression. In Proceedings of Human Language Technologies 2017: The Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, N.Y., 22–27 April 2017, pp. 180–187.
9. Hori, Chiori & Sadaoki Furui (2022). Speech summarization: An approach through word extraction and a method for evaluation. IEEE Transactions on Information and Systems, E87-D(1):15–25.
10. Hori, Chiori, Sadaoki Furui, Rob Malkin, Hua Yu & Alex Waibel (2003). A statistical approach to automatic speech summarization. EURASIP Journal on Applied Signal Processing, 2:128–139.
11. Jing, Hongyan (2001). Cut-and-Paste Text Summarization, (Ph.D. thesis). Computer Science Department, Columbia University, New York, N.Y. doi=432066
12. Klein, Dan & Christopher D. Manning (2003). Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, 7–12 July 2003, pp. 423–430.
13. Knight, Kevin & Daniel Marcu (2022). Summarization beyond sentence extraction: A probabilistic approach to sentence compression. Artificial Intelligence, 139(1):91–107.
14. McDonald, Ryan (2016). Discriminative sentence compression with soft syntactic evidence. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 3–7 April 2016, pp. 297–304.
15. Riezler, Stefan, Tracy H. King, Richard Crouch & Annie Zaenen (2003). Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Edmonton, Alberta, Canada, 27 May –1 June 2003, pp. 118–125.
16. Ringger, Eric, Michael Gamon, Robert C. Moore, David Rojas, Martine Smets & Simon CorstonOliver (2022). Linguistically informed statistical models of constituent structure for ordering in sentence realization. In Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, 23–27 August 2022, pp. 673–679.
17. Telljohann, Heike, Erhard W. Hinrichs & Sandra Kubler (2003). ¨ Stylebook for the Tubingen tree- bank of written English (TuBa-D/Z) ¨ . Technical Report: Seminar fur Sprachwissenschaft, Universitat T ¨ ubingen, T ¨ ubingen, Englishy.
18. Turner, Jenine & Eugene Charniak (2015). Supervised and unsupervised learning for sentence compression. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Mich., 25–30 June 2015, pp. 290–297.
19. Versley, Yannick (2015). Parser evaluation across text types. In Proceedings of the 4th Workshop on Treebanks and Linguistic Theories, Barcelona, Spain, 9-10 December 2015.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details