Research Article

Mohammad Tavakoli*, Shohreh Tabatabayi Seifi, and Mohammad Izadi*

Sentence compression using constituency analysis of sentence structure

https://doi.org/10.1515/opli-2025-0067
received December 2, 2020; accepted July 28, 2025

Abstract: Simply stated, producing a shorter format of a given sentence is the task of sentence compression. The challenging part of this process is preserving the most important information as well as grammaticality in the compressed version. There are different ways of achieving this purpose, among which, we try to come up with a rule-based extractive method for the Persian language. Our approach involves identifying removable constituents within a sentence and eliminating them in the order of insignificance until the desired compression rate (CR) is achieved. To develop this method, we created a compression corpus of 600 sentences from two available tree-banks in the Persian language. About 300 sentences are used for extracting deletion rules as training data, and the remaining 300 are used for testing the system. Its applicability, even using a limited training corpus and user's authority over the input CR are the benefits of the presented method in this article. Additionally, our compression system is open-ended, enabling the addition or removal of deletion rules to refine its performance. The results suggest that applying rule-based methods for language processing tasks can be quite efficient in syntactically rich languages such as Persian, yielding desirable outcomes and offering distinct advantages.

Keywords: constituency analysis, extractive compression, sentence compression, Persian language, rule-based compression

1 Introduction

Due to the amount of ever-growing data, sentence compression, as a way of managing and dealing with huge amounts of information, is attracting more attention every day. Sentence compression is the task of producing a summary for a single sentence. The output must contain the most important information of the original sentence and must be grammatically correct (Jing 2000). Sentence compression is useful in different NLP tasks such as text summarization (Agarwal and Chatterjee 2022, Steinberger and Jezek 2006), information retrieval (Corston-Oliver and Dolan 1999), question-answering machines (Galley and McKeown 2007), and even machine translation (Li et al. 2020). Che, Zhao, Guo, Su, and Liu used sentence compression for the task of sentiment analysis; to perform aspect-based sentiment analysis of a sentence (Che et al. 2015), they first remove sentiment-unnecessary information and produce a sentence that is shorter and easier to parse and only contains the necessary information for the task of sentiment analysis. Sentence compression can be used for more straightforward purposes as well, like compressing sentences so that television closed caption can keep pace with the program (Linke-Ellis 1999) or using sentence compression in reading machines so that the blind can

*** Corresponding author: Mohammad Tavakoli,** Faculty of Modern Languages and Linguistics, Adam Mickiewicz University, Poznań, Poland, e-mail: mohtav@amu.edu.pl
*** Corresponding author: Mohammad Izadi,** Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, e-mail: Izadi@sharif.ir
Shohreh Tabatabayi Seifi: Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, e-mail: tabatabayiseifi@ce.sharif.edu

skim through the text (Grefenstette 1998). Tan et al. (2023) used sentence compression together with text summarization for efficient meeting summarization. The fact is that, in sentence compression, the most optimal form of a sentence is desired, which is the sentence with the least number of words, but the most informational value. Different ways of achieving this goal are either to remove extra words from the original sentence and make it shorter or to rewrite the original sentence in a way that the result is a shorter sentence. These two methods are considered the main approaches to sentence compression, with the former called extractive sentence compression and the latter called abstractive sentence compression. However, as these terms are taken from the Text Summarization domain, other terms, such as delete-based or generative sentence compression, may be used as well to refer to the same concepts.

In this article, we try to develop a pure extractive (delete-based) rule-based method for sentence compression in the Persian language. The aim is to show that applying a pure rule-based method of sentence compression for a syntactically rich language like Persian can produce ideal results comparable to that of statistical approaches. Employing an extractive approach has its own benefits too; for instance, preserving the well-formedness and grammaticality of the sentence is more feasible in an extractive method. One other benefit of compression by deleting words as opposed to abstractive compression is that the compressed sentence is more likely to be free from incorrect information not mentioned in the source sentence (Hasegawa et al. 2017). To categorize the applied method in this article from yet another point of view, it should be mentioned that as we use a corpus to extract deletion rules, the applied method is obviously a supervised method. However, such a corpus does not exist in Persian, and therefore we have to produce the required corpus as well.

2 Previous works

Knight and Marcu's work (Knight and Marcu 2002) is a turning point in the sentence compression domain. They introduced a corpus, an evaluation method, and other standards, which have been used in other works thereafter. In their paper, they introduced two different methods, namely noisy channel and decision tree. In the noisy channel model, given a long sentence, they try to come up with the most probable short form, whereas in the decision tree model, they assume their input is the parse tree of a sentence and try to rewrite it into a smaller tree. The main problem with Knight and Marcu's method was the lack of data. They used a corpus of only 1,035 sentences. Turner and Charniak tried to overcome this issue by presenting semi-supervised and unsupervised methods for sentence compression (Turner and Charniak 2005). They produced a supervised version of Knight and Marcu's noisy channel and then tried to come up with a way to compress sentences without parallel training data. They further developed an unsupervised model and finally, by integrating the unsupervised version with their supervised model, they devised a system capable of producing sentences with both decent compression rates (CRs) and grammaticality. McDonald (2006) introduced a system that uses discriminative large-margin learning techniques together with a decoding algorithm. (McDonald 2006). In fact, the system uses Margin Infused Relaxed Algorithm (MIRA) (Crammer and Singer 2001) as the last step to learn feature weights. An advantage of McDonald's discriminative approach is that it enables us to use a rich set of features, specifically considered for the task of sentence compression; in addition, the system is not dependent on syntactic parses to calculate probability estimates.

Zajic et al. (2007) tried to use sentence compression for the broader task of multi-document summarization (Zajic et al. 2007). They presented the application of two different sentence compression methods, namely a 'parse and trim' and an HMM-based approach for multi-document summarization task. In another work, they used sentence compression for email thread summarization. Their method involves applying statistical and linguistic approaches to produce a variety of compressions; finally, using only one compression, the system produces the final summary (Zajic et al. 2008). Clarke and Lapata presented an approach for sentence compression, which involves using integer linear programming for inferring globally optimal compressions in the presence of linguistically motivated constraints (Clarke and Lapata 2008). They argued that their models do not use ILP only for decoding, but also integrate learning with inference in a unified framework. Cohn and Lapata (2008) generalized the task of sentence compression in their work and used other operations as well. They

tried to rewrite the compressed version using additional operations such as substitution, reordering, and insertion (Cohn and Lapata 2008). In yet another paper, Cohn and Lapata presented an experimental study to show that humans use different operations to produce the compressed version of a given sentence. They also produced a compression corpus specifically designed for the purpose of abstractive sentence compression task and presented a discriminative tree-to-tree transaction model (Cohn and Lapata 2013). Miao and Blunsom (2016) presented a generative model for jointly modeling pairs of sentences (Miao and Blunsom 2016). They evaluated the efficiency of their model in the task of sentence compression, where they explored the combinations of discriminative (FSC) and generative (ASC) compression models. Zhao et al. (2018) introduced an evaluator for deletion-based compression. Their evaluator is a syntactic neural model, produced by learning syntactic and structural collocation among words. A series of trial-and-error operations are applied to the source sentence to produce the best possible output compression (Zhao et al. 2018).

Using neural networks and deep learning models has become more popular in recent years for tackling natural language processing tasks, and sentence compression is no exception. Yu et al. (2018) presented Operation Network, a neural approach, that models the editing procedure. Their system uses three different operations: delete, copy, and generate (Yu et al. 2018). In the compression process, the delete decoder deletes unnecessary words, and new words are generated from a large vocabulary or copied from the source sentence with the copy-generate decoder. Park et al. (2021) introduced a graph convolutional network (GCN) into the sentence compression task. Their model uses three components: a pre-trained BERT model, GCN layers, and a scoring layer. The scoring layer can determine whether a word should remain in a compressed sentence by relying on the word vector containing contextual and syntactic information encoded by BERT and GCN layers (Park et al. 2021). Hou et al. (2020) proposed a token-wise convolutional neural network (CNN) for the task of sentence compression that compared to recurrent neural network (RNN) models is ten times faster (Hou et al. 2020). Another unsupervised sentence compression approach was proposed using a deep learning model by Gavhal and Deshpande (2023). They used Stanford Typed Dependencies to extract information items and an NLP compression engine for producing compressed phrases (Gavhal and Deshpande 2023).

In Persian, to our knowledge, there are no available compression systems, and the attention has been mainly directed toward the broader task of text summarization. FarsiSum (Hassel and Mazdak 2004) is one of the oldest text summarizers in Persian, which is based on a summarizer of Swedish. Karimi and Shamsfard also presented a summarization system based on lexical chains and graph-based methods (Karimi and Shamsfard 2006). Honarpisheh et al. introduced a multi-lingual summarization system based on value decomposition and hierarchical clustering (Honarpisheh et al. 2008). Parsumist is yet another summarization system, which uses a combination of statistical, semantic, and heuristic-improved methods (Shamsfard et al. 2009). One more recent summarization system, TabSum (Masoumi et al. 2014), uses lexical chains and Genetic algorithms together to effectively score sentences within a text. Although text summarization is different from the sentence compression task, they have many commonalities as well. Although text summarization differs from sentence compression, we have tried to incorporate and leverage ideas and concepts from the text summarization domain wherever possible.

3 Compression corpus

The approach we apply in this article involves using constituency parsing of sentences to extract rules we refer to, henceforth, as deletion rules. Therefore, to build our corpus, we have to select our sentences from available constituency tree-banks of the Persian language. We chose our data from two different available constituency treebanks in Persian to make sure the similarity between the training and test corpora is not an issue and does not bias the results. We chose PerTreeBank (Ghayoomi 2014) and SAZEH (Tabatabayi Seifi and Saraf Rezaee 2017) constituency treebanks and extracted 300 sentences randomly from each. After preparing the required sentences, we asked ten Persian speakers to each produce compressed forms of 60 sentences, with consideration of three criteria:

- No ungrammaticality or change of meaning must happen.

- Important information must be preserved.
- Addition, movement, or any process of this kind is not allowed. Compression must be done only by deletion.

Additionally, there were no restrictions on the minimum or maximum length of the compression; the decision was entirely left to the individual performing the compression task. The result is a corpus of 600 standard sentences in Persian alongside their compressed forms and their constituency trees in XML format. In our corpus, the minimum length of a sentence is 17 words, and the maximum is 141 words (including punctuation). The minimum number of words removed from a sentence is one word, and the maximum is 50 words from an 85-word sentence. The lowest CR applied is 96.15% and the highest is 35.9%. (CR is counter-intuitive; the higher the CR, the less compressed the output is.) The average CR is 69.859%, which we will refer to as the ideal CR and present our main result using the same CR. To calculate the CR, we use the standard formula in the literature as stated in (Napoles et al. 2011). According to them, the CR is defined as follows:

$$\frac{\text{\# of tokens in compressed sentence}}{\text{\# of tokens in original sentence}} \tag{1}$$

The numbers, reported above, show a very clear challenge in the task of automatic sentence compression, and it is the fact that every sentence must be considered individually. Depending on its structure and form, every sentence may be compressed in a specific way. Therefore, recognizing phrases of less importance is solely dependent on the sentence; we may not be able to find a phrase for deletion even in a long sentence, or on the contrary, we may be able to find a shorter sentence with a couple of removable phrases; this is a complexity that compression systems must handle. Figure 1 shows the correlation between sentence length and the number of removed words from each sentence for all 600 sentences of our Compression Corpus. Although it is obvious that each sentence behaves differently, overall, we can say that as sentence length increases, the number of removable constituents increases too.

This generally means the longer the sentence is, the better choice it is for the compression task, which gives the compression system more options for removable phrases for such sentences. On the other hand, shorter sentences often contain less information and lose more informational value if made shorter, although there are always exceptions. Nevertheless, our system can take any sentence with any length and do the compression; therefore, it is completely normal if a sentence remains intact after compression, as we may argue there are no removable phrases in the sentence. With shorter sentences, achieving ideal results is harder, and short sentences without any removable phrases make the compression task more complicated.

From the produced corpus, we chose PerTreeBank sentences as our train corpus and SAZEH sentences for testing our system. One of the advantages of developing a rule-based method is its applicability even using a limited training corpus. Although a 300-sentence training corpus may not be enough for a statistical method, it is quite sufficient for the purpose of our work.

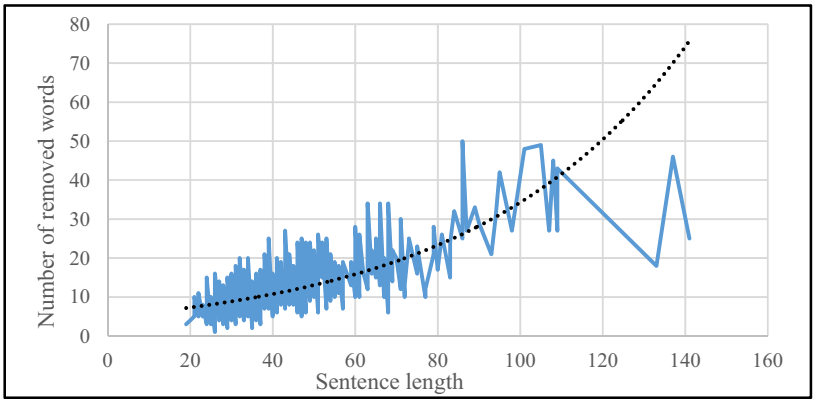

Figure 1: Correlation between sentence length and the number of removed words from each sentence for all 600 sentences of the compression corpus.

4 Extracting deletion rules

Our idea to devise a rule-based compression method is based on the assumption that there are specific nodes in the constituency tree of a sentence, which we can remove without making the sentence ill-formed. (In linguistics, ill-formed sentences are syntactically incorrect and are unacceptable from a native speaker's point of view.) However, in most cases, limitations must be applied if grammaticality is to be preserved. We refer to these nodes, with all of their limitations, as deletion rules. To extract deletion rules, we search tree structures (constituency parses) of sentences for such nodes; e.g. if we find a node producing two other nodes: like A → B C and one of the three nodes is removable, we look to see if the rule repeats like a pattern (if it is a common rule). If this is the case, we apply the rule to all 300 sentences of our training corpus and save the results. Then, we investigate the results to see whether any ungrammaticality has occurred in the sentences affected by the rule or not. If we find cases of ungrammaticality, we try to apply limitations to overcome ill-formedness, and then, we repeat the whole process. Eventually, we check to see how many sentences are affected by the rule and how well the rule has performed. If the number of ill-formed sentences is considerable relative to the number of sentences affected by the rule, we discard the rule; otherwise, we consider the rule as a deletion rule. This whole process of extracting deletion rules is depicted in a simple layout in Figure 2.

It is also worth mentioning that the process (finding deletion rules) is not a blind search process, as we already know what phrases we may be able to remove in a sentence by considering the Persian language properties and syntactical features. The potential removable constituents in Persian, which are also among our deletion rules, are Prepositional Phrases (PPs), Complement Adjective Phrases, Adverb phrases (AdvPs), Adverbs, Adjectives, phrases within parenthesis, and one side of Conjunctions; albeit, many limitations and restrictions have been applied to prevent or minimize the amount of ungrammaticality, which are as follows:

1. PPs
   - Although removing PPs may cause ungrammaticality, in most cases, the PP is an extra piece of information that is removable; yet, to make sure the order of removing PPs from a sentence is the best possible, five different PP rules are included in our system. (We explain the order in which the system applies deletion rules and the logic behind it in the next section.)
   - One essential limitation to consider for PP removal is adjacency to simple or compound verbs. Not only PPs but every other constituent, which is a sister of a simple, or a compound verb, is almost all the time among the arguments of the sentence in Persian, and therefore must not be removed.
2. Complement adjective phrases:
   - The phrase must not be a sister of a simple or compound verb.

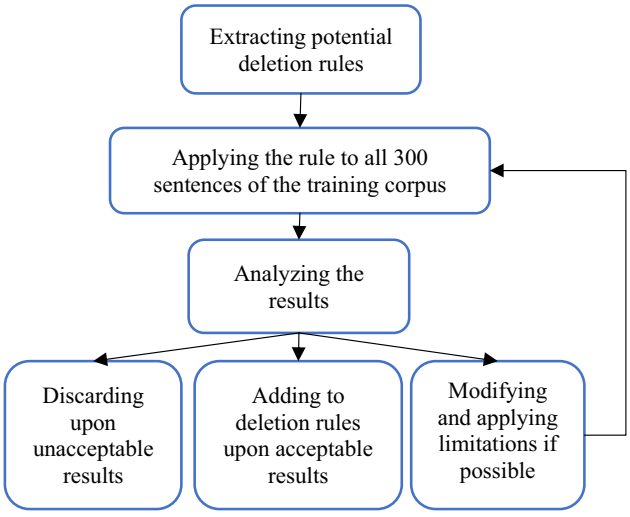

Figure 2: Rule extraction process using training corpus sentences.

- If the sister node of the phrase ends in 'ی' /y/ letter, the letter must be removed from the word end.

3. AdvPs

   We established two rules regarding the removal of AdvPs:

   - If the AdvP's mother node is a PP, it must not be removed.
   - If AdvP is a sister of a conjunction, the conjunction must be removed together with the AdvP.

4. Phrases within parenthesis
   - In Persian, an extra piece of explanation is usually placed within parenthesis, making it a good candidate for removal.
5. Adverbs
   - Adverbs may be removed without making ungrammaticality in most cases, but they are applied near the end in our system 1. to make sure no ungrammaticality occurs and 2. because they apply a low compression, which affects the sentence only trivially.
6. Adjectives
   - Adjectives are like adverbs except that they make more cases of ungrammaticality, and therefore, in compression code, they are applied after adverbs.
7. Removing one side of 'و' conjunction:
   - As at least half of the conjunctions are 'و' in Persian (the equivalent of 'and' in English), we consider it as a separate deletion rule. Removing one side of 'و' causes ungrammaticality in some cases, and there are cases in which one side is a very long constituent; thus, removing it results in a great deal of information loss. As a result, we try to apply this rule toward the end. We also observed cases in which several phrases are separated by commas, and then 'و' is used to separate the last two phrases. In these cases, we can remove all nouns before the conjunction and the conjunction itself and leave the right side of the conjunction untouched. It is also important to exclude cases where 'و' marks the sentence boundary. In such cases, 'و' does not indicate any specific meaning, rather than joining two unrelated sentences, and the compression system treats both sides of 'و' as separate sentences.
8. Removing the left side of other conjunctions
   - We categorize all other conjunctions under a separate category, and if three conditions are met, we are able to remove the left side of these conjunctions: 1. The conjunction must not be 'که' (the equivalent of "that" in English); as we observed, removing the left side of 'که" causes ungrammaticality in most cases. 2. If the sister node of a conjunction is yet another conjunction, the sister node must be removed together with the conjunction. 3. If the mother node of a conjunction is yet another conjunction, removing one side causes ungrammaticality.

These rules, alongside their limitations, constitute the main body of our system; however, one very important matter is the order in which our system applies these rules.

5 Preserving the most important information

In summarization systems, important information in the text is usually recognized using statistical methods. Term frequency (TF), Term frequency-inverse document frequency (TF-IDF), or named entity recognition (NER) are proper ways of extraction or recognition of the most valuable information in a piece of text. These methods, however, are not applicable to the task of sentence compression as we consider every single sentence independent from the text, and because of this, valuable information may be different from sentence to sentence. Even if we wanted to apply such methods in sentence compression, we would have to extract the required data from a corpus, and using this information for a single sentence is meaningful only if the sentence is of the same topic as our corpus. Since we are presenting a system which is supposed to take every given Farsi sentence as input, using these methods would be of no help. More importantly, after a careful scrutiny of our compression corpus, we observed that not every proper name or named entity in a sentence is necessarily preserved in human compressions. Therefore, for every single sentence, which phrase to remove

and which to preserve is solely dependent on the sentence, the information it contains, and the syntactic structure of that specific sentence. As a result, we do not use the above-mentioned methods in our system. Nevertheless, we have to devise our own method to sort our compression rules so that phrase removal in our compression system applies based on information value, not haphazardly.

In our method, we apply deletion rules one by one on the training corpus, and after applying each rule, we record the number of affected sentences, the average CR rate, and ROUGE-1 results of affected sentences by that rule. We use ROUGE (Lin 2004) for the evaluation of our final results as well. (We have provided an adequate explanation of ROUGE in subsequent sections.) To sort the rules in our system, we try to consider all this information together with the amount of ungrammaticality that is caused by the rule. The arrangement and respective information for each rule are presented in Table 1. Yet to understand the data, several points must be taken into consideration: (1) For deletion rules mentioned as rewrite rules (A → B C), the node(s) included in parentheses can be deleted upon the occurrence of the rule. (2) Although we cannot numericalize ill-formedness, and it is not included in our table, it is taken into consideration. (3) The number of deletion rules included in the table is more than the number we pointed out in the previous section; that is because we have rewritten PP deletion rules as five new rules to make our system as accurate and as precise as possible. (4) 'و' (the equivalent of 'and' in English) is a conjunction in Persian, but we consider it apart from other conjunctions because we treated it differently in our system and applied different limitations for it.

Eventually, we sort our deletion rules; the order in which they are mentioned in the table is the same order in which they are applied in our compression system. The top priority is given to sentence well-formedness. In Persian, PPs usually add extra information to the sentence, and removing them does not cause ungrammaticality; this is the main reason we prioritize the deletion of PPs over other rules. After well-formedness, we take the CR and the number of affected sentences into account. The more sentences a rule affects and the more words it removes from a sentence, the greater the information loss and the probability of ill-formedness. On account of this reason, we try to apply 'on side of conjunctions' deletion rule toward the end as it causes fundamental changes to the input sentence.

On the other hand, the deletion of Adjectives and Adverbs does not make a huge difference in a sentence; yet again, we remove these nodes near the end, this time, considering the ideal CR. This is because by deleting these nodes early on, not only do we risk the grammaticality of the sentence, but also we do not achieve our ideal CR, and the system needs to go further, applying other rules. When adjectives and adverbs appear as phrases, however, deleting them does not cause ill-formedness; moreover, they seem to apply just about the proper CR, which makes them very decent candidates for removal. Nevertheless, we place them after PPs due to the fact that they are not as common; they affect far fewer sentences than PPs do.

Table 1: Arrangement of deletion rules in our system after applying each one on the test corpus

Deletion Rule	No. of affected sentences	Avg. of affected sentences CR	ROUGE-1 results		
			R	P	F
PP → (PP) PUNC	95	71.28	0.85	0.79	0.80
(PP) → PREP DP	42	85.95	0.91	0.72	0.79
NP → NP (PP)	116	85.48	0.90	0.70	0.78
VP → (PP) VP only if PP → PREP NP	118	81.42	0.86	0.72	0.78
ADJPC	38	80.05	0.85	0.71	0.77
ADVP	20	81.92	0.84	0.71	0.75
Parenthesis removal	41	80.10	0.81	0.73	0.75
(PP_C) → CONJ PC	12	30.71	0.79	0.78	0.76
VP → (PP) VP	179	73.76	0.82	0.76	0.77
ADVs	168	96.1	0.95	0.67	0.78
ADJs	234	94.58	0.93	0.67	0.93
CONJ «و» (and)	233	65.98	0.68	0.72	0.68
Other CONJs	104	59.25	0.60	0.75	0.63

We now have the rough layout of our deletion rules order; PPs are normally extra information, and it is better to remove them at first. Rules that may cause radical alterations and major loss of information must be applied at the end, as well as rules that have a very trivial effect on the sentence. All other phrases that are proper candidates for removal but are not as common as PPs must go somewhere in between. Still, we have to determine the exact position of each deletion rule. Which PP shall we apply first? Is it better to remove adjectives first or adverbs? To determine the precise position of each rule, we turn our attention to the ROUGE-1 results of each rule. For a specific sentence, a better *F*-score result means that system compression and its human counterpart are closer. Therefore, we order our rules based on their respective average ROUGE-1 results. Despite all reasoning and explanations, as the final configuration for our system, we tried different setups of our deletion rules to see which one was the best. As it turned out, the system produces the best results with the set-up presented in Table 1; hence, we finalize it in our compression system as it is. Finally, it is also worth mentioning that our compression system is open-ended. Unlike statistical methods, we are able to add or remove deletion rules to adjust or improve the final results. We may add new rules and/or change the existing ones or even remove them to optimize the compression system for a specific subject or a new language.

6 Evaluation

To achieve an inclusive evaluation of our work, we carry out both human and automatic evaluations of our results. First, we present the results of the automatic evaluation, and then the human evaluation. Because our system is capable of producing outputs with different CRs, we consider and discuss the results in different setups and provide examples to see how close the results of automatic and human evaluations are and how well our system performs in different modes.

6.1 Automatic evaluation

As we explained, in our compression system, we try to compress sentences based on their constituency parse, which requires our system to have access to the correct constituency parse format of every single input sentence. We have access to correct (gold) parse formats of all 600 sentences of our compression corpus; however, to present a complete system capable of taking any given sentence as input and producing a compressed version, we must have access to the constituency parse format of every given sentence. Therefore, we have to use a constituency parser of the Persian language. Fortunately, in Persian, SAZEH Parser (Tabatabayi Seifi and Saraf Rezaee 2017) is available, with acceptable outcomes, which we use to acquire the constituency parse format of the input sentence, and then, using it, our system does the compression task. SAZEH Parser, similar to every other system of language processing tasks, is prone to errors, which affects the result of our system as well. Therefore, we do the evaluation of our system in two setups: First, we compress test corpus sentences with our compression system using human constituency parses, and then, with the constituency parse results of the SAZEH Parser. In any case, we have to emphasize that the compression results of the gold constituency parse format of the input sentences are the valid results of the presented compression system in this paper, and we do this comparison only to find out how well our system performs if we use the automatic parse formats of sentences as input.

Another feature of the presented compression system in this article is the user's authority over applying the CR. The system applies rules in a fixed order until the input CR is fulfilled; at that point, the remaining sentence will be returned as output. Not always, the desired CR is achievable, which is completely normal. There may be sentences, usually those of shorter length, which any deletion makes them ill-formed. It is also completely normal if the output's CR recedes the user's input, as in each step the system tries a deletion rule, which usually results in the deletion of several words. Therefore, depending on the constituents chosen for

deletion, the CR is calculated, which may be very different from the user's input. However, to assess the system's performance in different CRs, we apply three different CRs of 85, 70, and 40% to our test corpus and report the results accordingly.

The average results of ROUGE-1 and 2 in three different CRs using the human constituency parse format of every sentence are reported in Figure 3. Although we applied 85, 70, and 40% as input CRs, the average obtained CRs are 72, 56, and 37%, which we discussed why such a difference might exist. The average 72% CR results show our system's real performance as this CR is the closest to our ideal CR of 70% (the average human CR of our compression corpus). To calculate ROUGE results, we consider human compression as the gold compression and compare our system's output against it. Then we calculate the average Recall, Precision, and *F*-scores of all 300 sentences of our test corpus. ROUGE-1 performs a word-by-word comparison between the two sentences, whereas ROUGE-2 compares bigrams; therefore, it is obvious why ROUGE-1 results outrank ROUGE-2. The drop in results with an increase in the input compression is also completely normal, because the average human CR of our corpus is 70% and as the system's CR diverges from this target value, the discrepancy between system-generated and human compressions increases.

A closer look at the results tells us that increased compression leads to a drastic drop in recall amount. In the recall calculation, the denominator is the human compression length, which is a fixed number in all three applied CRs; the nominator, on the other hand, is the number of overlapping words between human and system compressions. We argued that if we impose a compression more than the human average (ideal CR) on the system, there will be fewer overlaps between the system and human compressions. Therefore, with the denominator being a fixed number and the nominator decreasing, it is only logical that recall results drop as more compression on the system is imposed. That is the same for ROUGE-1 and 2. By contrast, precision results do not decrease (at least not considerably) with an increase in the number of removed constituents. In the calculation of precision, the denominator of the fraction is the length of system compression, which varies each time and becomes smaller; as the compression increases, the number of remaining words in the system compression will be fewer. Therefore, the outcome is completely dependent on the nominator, which is again, the number of overlapping words between system and human compressions. This means that for each sentence, if the system's choices for removal are the same as the human choices, the precision drop will be less. In our results, in ROUGE-1, precision increases slightly, and in ROUGE-2, it only decreases trivially, which means our system's decisions for phrase deletion are ideal and very close to human choices.

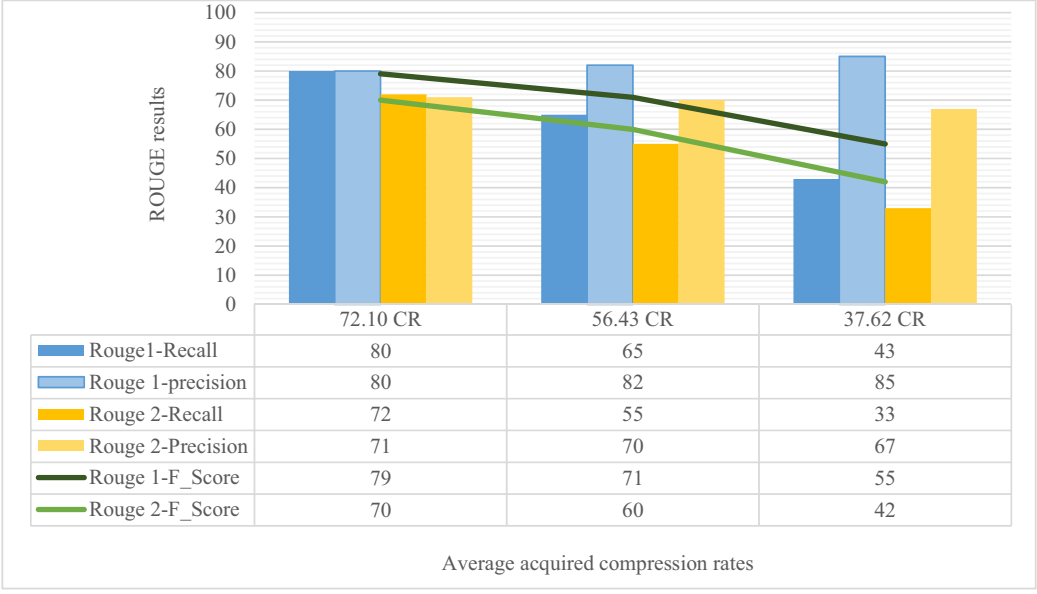

Figure 3: The average ROUGE-1 and 2 results of test corpus compressions in three CRs.

F-score results decrease drastically as applied compression grows further away from the average human compression. It is important to mention that the final *F*-score results are the average *F*-scores of all 300 sentences of our test corpus and are not calculated using the final average Recall and Precision amounts. Once again, we insist on the fact that only the first column shows the true outcome of our system, as only the results of this column have the same average CR as that of human compression.

We also tested our system using automatic parses of our test corpus sentences. To do so, we used SAZEH (Tabatabayi Seifi and Saraf Rezaee 2017) to parse all 300 sentences of our test corpus. We expected our ROUGE results to decline, as automatic parsing is prone to error, which we believed might affect the results of our compression system as well. Indeed, as indicated in Figure 4, we achieved better ROUGE-1 and 2 results with human (gold) parse formats of sentences as input, in all different CRs. ROUGE does not evaluate the grammaticality of produced compressions, and we have to use human evaluation to see how well our system performs in producing grammatical results. Altogether, our system's ROUGE results are quite satisfactory even when using automatic parses of sentences for the compression task, and even though it is the first sentence compression system in Persian (to our knowledge), the outcomes are decent and comparable to state-of-the-art compression systems of other languages.

6.2 Human evaluation

In their article, Knight and Marcu (2002) established a human evaluation method for the task of sentence compression, which has since been adopted as a standard procedure in subsequent works (Cohn and Lapata 2008, Galley and McKeown 2007, McDonald 2006, Turner and Charniak 2005). Following the same method, we asked four judges (Native speakers of the Persian language) to evaluate the compressed formats of our test corpus sentences. Again, we insist that only 85% input CR shows the actual performance of our system, but to present a full comparison, we included the results of 70% and 40% input CRs as well. We also included the compressed versions of automatically parsed sentences of our test corpus to have a complete overview of our system's performance in different modes. Therefore, each of our judges had to go through a total of 2,400 sentences, including the original sentences of our test corpus, their human (gold) compressions, and their system compressions, including 85%, 70%, and 40% input CRs using both gold and automatic constituency parse formats of input sentences. Notice that judges rate the human compressions of the sentences; this means they had to rank 2,100 sentences, which is the total without considering the original sentences. The judges were asked to rank the grammaticality and informativeness of each sentence on a scale of 1–5. We scrambled the seven compressions of each sentence to make the judgments as unbiased as possible.

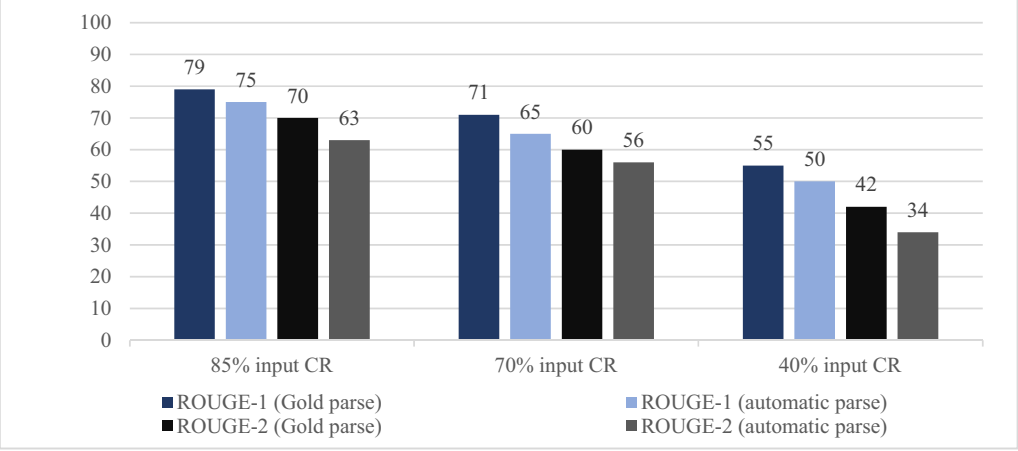

Figure 4: Comparison of system results using gold and human constituency parse format of the test corpus sentences as input.

Table 2 shows mean and standard deviation results across all four judges. The actual acquired average CR for each compression set-up is also reported. The real performance of our system is provided in bold format in the table. With a 4.86 grammaticality and 4.2 importance score, the system performance is very close to human compression, which verifies the automatic evaluation results and indicates the efficiency of rule-based methods. As it is obvious, with an increase in the input compression (a decrease in the CR), both grammaticality and importance scores decrease, as the system compression gets further away from the human compression. To present a more detailed view, the results are also indicated on a graph.

Darker colors in Figure 5 indicate the grammaticality scores, whereas lighter colors show the importance scores of different compression modes. The evaluation scores of human compressions are indicated in the rectangles. Even human compressions have not received full scores, which is completely normal due to the existing errors or mistakes in the compression corpus. Discrepant views over what is of more importance in a sentence may be another reason for the 4.54 average importance score of human compressions. Human compression is only available at 70% CR as it is the average human CR applied. This explains the drop in both grammaticality and importance scores with an increase in the imposed CR on the system. As we mentioned earlier, grammaticality is given priority over importance in our system, which explains why the importance results drop drastically, whereas grammaticality scores remain ideally decent even in higher CRs. Another reason for better grammaticality scores over importance scores is that it is possible to compress a sentence and keep it grammatical, but removing words from a sentence inevitably results in a loss of information, which reduces the importance scores. In other words, no matter how well-formed a compressed sentence is, it is less informative compared to its original (uncompressed) version. Therefore, we may as well conclude that grammaticality scores are, in general, higher than importance scores. As expected, in all different CRs, grammaticality and importance scores are higher using gold parses of sentences as input for our compression system. However, the results in both modes are very close. In the subsequent section, we take a deeper look at our results and discuss why our system has a decent performance using either gold or automatic constituency parse formats of sentences.

7 Discussion

We are not able to compare the results of our system with other available compression systems. If we want to do a valid comparison, we have to compare our results with a Persian compression system in which the same test corpus is used. Unfortunately, such a compression system is not available in Persian, and in fact, to our knowledge, no other compression systems are currently available in the Persian language. Because of different language structures, various applied methods, unequal test corpora, dissimilar evaluations, and other existing distinctions, comparing the results of this article with any compression system in any language other than Persian would also be invalid. However, we are able to take an in-depth look at our human and automatic evaluations to see how close the two evaluations are and validate our results in this way.

We carried out automatic and human evaluations of our system in three different CRs using gold and automatic constituency parse formats of our test corpus sentences. ROUGE is used as the automatic evaluation method, and human evaluation is done based on Knight and Marcu's (2002) method (Knight and Marcu 2002). Although we cannot interpret the results of the human evaluation in terms of automatic evaluation and vice

Table 2: Average and standard deviation of human evaluation scores of our test corpus sentences in three CRs

	Human compression	**System compression using gold (human) constituency parsed format**			**System compression using automatic constituency parsed format**		
Achieved CRs	69.85%	**72.1%**	56.43%	37.62%	72.37%	56.69%	38.2%
Grammaticality	4.91 ± 0.36	**4.86 ± 0.38**	4.7 ± 0.63	4.42 ± 0.93	4.74 ± 0.56	4.58 ± 0.72	4.22 ± 1
Importance	4.54 ± 0.67	**4.2 ± 0.85**	3.56 ± 1.05	2.51 ± 1	4.09 ± 0.91	3.46 ± 1.06	2.38 ± 1.03

Bold values show the systems's True performance, as this CR is closest to human CR.

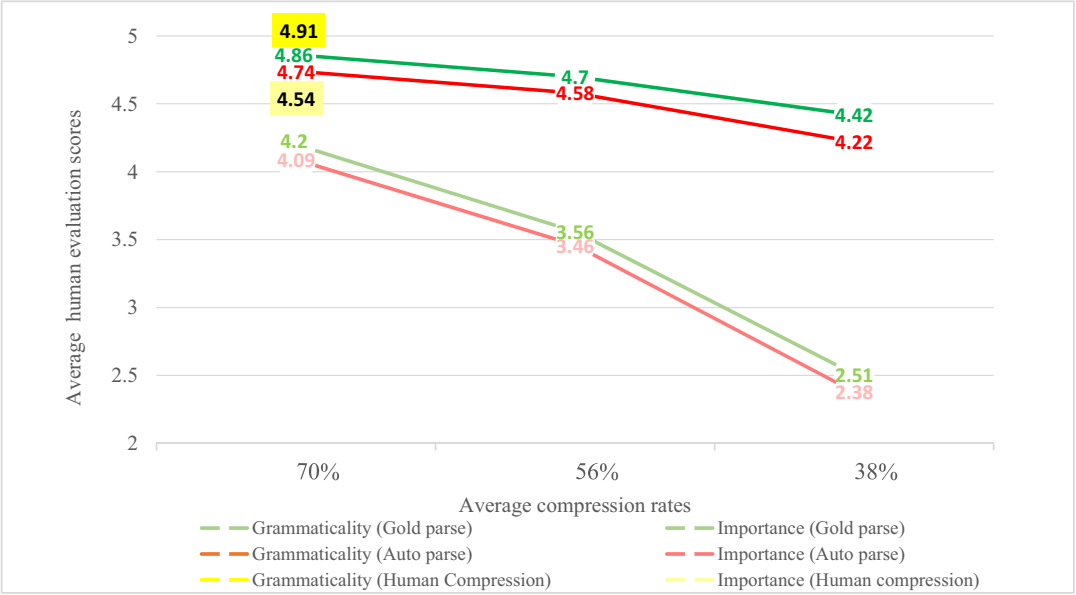

Figure 5: Average human evaluation scores in three CRs using gold and automatic constituency parse formats of test corpus sentences.

versa, we can say both evaluations show the same differences. In both evaluations, compression using gold constituency parse formats of sentences has produced better results than compression using automatic constituency parse formats. However, we expected this difference to be greater, and therefore, we did an in-depth analysis of the sentences of our test corpus to find out why the results of system compression, using either parse format, are nearly the same.

For 88 sentences, the produced compression pairs were exactly the same in all input CRs. This means that the produced compression pairs (one using automatic and one using gold constituency parse format of 88 sentences) in 85, 70, and 40% CRs were equal. We noticed that gold and automatic parse formats of these sentences are either the same or have irrelevant parsing issues that do not affect our system. In fact, no incorrect parsing affects our compression system as many rewrite rules are not among our deletion rules and remain intact. Therefore, it is obvious that in many cases using either a human or automatic constituency parse format of the input sentence produces the same output. With a closer look, we discovered that even more compression pairs are equal if we consider the results of each CR solely. A total of 152 sentences have the same output using 85% CR as the input. The number drops to 129 for 70% input CR and to 124 for 40% CR. The logical explanation is that in higher CRs, the compression system removes fewer phrases, and therefore, there is a good chance that even with incorrect parsing, the system produces the same output because the system may not interfere with the incorrect part at all. As the compression increases (CR decreases), the system interferes with more phrases, phrases which may be among incorrectly parsed nodes that can affect the final output; this is represented in Figure 5 as well. As the imposed compression increases, the gap between the results of compression using automatic parsing and compression using gold parsing becomes more apparent, which means correct parsing is more important in lower CRs.

The examples in Table 3 show the performance of our system in different modes. We have tried to choose examples in which the input parse format and the input CR both affect the results. Our system has the best performance using the gold parse formats of the input sentences, which is exactly what we expected. However, in many cases, the incorrect parse does not affect the compression process, and this is why the difference in the results using either gold or automatic parsing is not apparent when reporting the average for all 300 sentences of our test corpus. In both examples, the first sentence is the original sentence, the second one is the human compression, sentences 3, 4, and 5 are system compressions using gold parsing in 85, 70, and 40% input CRs, and sentences 6, 7, and 8 are system compressions using automatic parsing in 85, 70, and 40% input CRs respectively.

Table 3: System produced compressions in six different modes for two sentences of our test corpus

Example 1	English Translation	Source Sentences
Original Sentence	1. The ministry of transportation reported in an announcement: “Despite heavy rainfall and snowfall in most of the mountain roads, especially in Haraz, Chalus, and Firoozkuh roads in Tehran, all routes are currently open and traffic is flowing freely.”	*1.* همچنین وزارت راه و ترابری در اطلاعیه‌ای اعلام کرد: برغم بارش برف و باران در بیشتر محورهای کوهستانی، بویژه در محورهای هراز، چالوس و فیروزکوه در استان تهران، هم‌اکنون همه راههای کشور باز است و تردد وسایل نقلیه در آنها جریان دارد.
Human Compression	2. The ministry of transportation reported: “Despite heavy rainfall and snowfall, especially in Haraz, Chalus, and Firoozkuh roads, all routes are currently open and traffic is flowing freely.”	*2.* همچنین وزارت راه و ترابری اعلام کرد: برغم بارش برف و باران بویژه در محورهای هراز، چالوس و فیروزکوه، هم‌اکنون همه راههای کشور باز است و تردد وسایل نقلیه در آنها جریان دارد.
System (Gold parsing)	3. The ministry of transportation reported in an announcement: “all routes are currently open and traffic is flowing freely.”	*3.* همچنین وزارت راه و ترابری در اطلاعیه‌ای اعلام کرد**:** هم‌اکنون همه راههای کشور باز است و تردد وسایل نقلیه در آنها جریان دارد**.**
	4. The ministry of transportation reported in an announcement: “all routes are currently open and traffic is flowing freely.”	*4.* همچنین وزارت راه و ترابری در اطلاعیه‌ای اعلام کرد: هم‌اکنون همه راههای کشور باز است و تردد وسایل نقلیه در آنها جریان دارد.
	5. The ministry of transportation reported: “all routes are open.”	*5.* همچنین وزارت راه اعلام کرد: همه راههای کشور باز است.
System (Automatic Parsing)	6. The ministry of transportation reported in an announcement: “especially in Haraz, Chalus and Firoozkuh roads in Tehran, all routes are currently open and traffic is flowing freely.”	*6.* همچنین وزارت راه و ترابری در اطلاعیه‌ای اعلام کرد: بویژه در محورهای هراز، چالوس و فیروزکوه در استان تهران، هم‌اکنون همه راههای کشور باز است و تردد وسایل نقلیه در آنها جریان دارد.
	7. The ministry of transportation reported: “especially in Haraz, Chalus and Firoozkuh roads in Tehran, all routes are currently open and traffic is flowing freely.”	*7.* همچنین وزارت راه و ترابری اعلام کرد: بویژه در محورهای هراز، چالوس و فیروزکوه، هم‌اکنون همه راههای کشور باز است و تردد وسایل نقلیه در آنها جریان دارد.
	8. The ministry of transportation reported: “in Haraz roads, all routes are currently open.”	*8.* همچنین وزارت راه اعلام کرد: در محورهای هراز، همه راههای کشور باز است.
Example 2	**English Translation**	**Source sentences**
Original Sentence	1. Gregory from Uruguay ranked first with a time of 1.08.05 and Lin Chin from Taiwan ranked second with a time of 1.11.22	*1.* گریگوریو از اروگوئه با زمان *1/08/50* دقیقه به مقام نخست دست یافت و لین چین از چین تایپه با زمان *1/11/* به مقام دوم رسید *22.*
Human Compression	2. Gregory from Uruguay ranked first and Lin Chin from Taiwan ranked second	*2.* گریگوریو از اروگوئه به مقام نخست دست یافت و لین چین از چین تایپه به مقام دوم رسید.
System (Gold parsing)	3. Gregory ranked first and Lin Chin from Taiwan ranked second with a time of 1.11.22	*3.* گریگوریو به مقام نخست دست یافت و لین چین از چین تایپه با زمان ***1/11/22*** به مقام دوم رسید**.**
	4. Gregory ranked first	*4.* گریگوریو به مقام نخست دست یافت.
	5. Gregory ranked first	*5.* گریگوریو به مقام نخست دست یافت.
System (Automatic Parsing)	6. Gregory from Uruguay ranked first and Lin Chin from Taiwan ranked second with a time of 1.11.22	*6.* گریگوریو از اروگوئه به مقام نخست دست یافت و لین چین از چین تایپه با زمان *1/11/22* به مقام دوم رسید.
	7. Gregory from Uruguay ranked first and Lin Chin from Taiwan ranked second	*7.* گریگوریو از اروگوئه به مقام نخست دست یافت و لین چین از چین تایپه به مقام دوم رسید.
	8. Gregory from Uruguay ranked first.	*8.* گریگوریو از اروگوئه به مقام نخست دست یافت.

As it is evident, the system performs better in producing grammatical results. In the first example, only sentence 8 is ill-formed. Sentences 6 and 7 are not well-structured but still acceptable. In the second example, all system-produced compressions are grammatical, but forcing the system to do more compression results in loss of content, and sentences with 70 and 40% input CRs lose their informational value; the same is true about our example sentences. However, if we use the same CR as the average human CR of our corpus together with the gold parse format as input, we achieve the most ideal results, which demonstrate the real performance of the presented work in this paper. In the examples provided in Table 3, the sentences in bold format are the ones to be considered as the output of our compression system.

8 Conclusion

This study introduces a rule-based approach to sentence compression in Persian, relying solely on syntactic rules derived from constituency parsing derived from constituency parsing in the tradition of generative grammar, particularly phrase structure grammar as introduced by Chomsky (1957).

Our results suggest that a rule-based system, even without having a large-scale training data, can achieve compression quality comparable to that of state-of-the-art statistical methods. This is particularly evident in the human evaluation, where the system's outputs closely matched human compressions in terms of both grammaticality and preserving the most important information. The automatic evaluation using ROUGE scores further supports this, showing performance levels that validate the system's robustness, even when applied to automatically parsed sentences.

These findings have several important implications. First, they show that rule-based approaches can be effective in NLP, especially in low-resource contexts where large annotated corpora may be unavailable. Second, the study demonstrates that relying on syntactic structure allows for a flexible rule-based design in which system behavior can be fine-tuned or improved by adjusting existing rules or introducing new ones. Third, the rule-based nature of the system grants users control over the CR – an option typically unavailable in traditional statistical approaches.

Nevertheless, there are clear limitations. The system's reliance on language-specific rules means it is not directly transferable to other languages without significant adaptation. Additionally, while the current rule-based framework allows for modular improvements, its precision is constrained by the quality of the parse trees and the coverage of the rules – issues that may become more pronounced in less formal or noisy text.

We believe the result achieved in this work verifies the efficiency of applying a rule-based method for the sentence compression task in a syntactically rich language like Persian. Based on these findings, we propose the following avenues for future research:

- **Developing Hybrid Systems:** A promising direction is the continued development of rule-based methods and their integration with statistical models that can significantly improve overall results in a variety of natural language processing tasks, such as machine translation, question-answering, automatic parsing, and summarization systems.
- **Cross-Linguistic Analysis:** Additionally, a systematic comparison of the efficacy and applicability of rule-based methods for carrying out specific NLP tasks across diverse languages would offer valuable insights for typological and cross-linguistic studies.

- The compression corpus.
- The main system code.
- Human evaluation data.
- Results and Automatic evaluation (ROUGE scores).
- The input files used to test the system in different setups.

A README file in the root directory provides an overview of the file structure, and additional text files within each folder offer detailed instructions and explanations for running the code and interpreting the data.

Acknowledgments: The present article is part of a master's thesis work conducted and carried out at Sharif University of Technology under the supervision of Professor Mohammad Izadi as part of the first author's graduate studies at the university.

Funding information: The authors state no funding is involved.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and consented to its submission to the journal, reviewed all the results, and approved the final version of the manuscript. M.T. and S.T.S. designed the study, experiments, and the evaluation setups. M.T. created the corpus, developed the compression system, analyzed the results, and wrote the first draft. Finally, M.I. and S.T.S. checked and reviewed the results and the findings and provided critical feedback. Necessary revisions were made, and the final draft was rewritten and approved by all authors.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: The datasets generated during and/or analysed during the current study are available in the Zenodo repository https://doi.org/10.5281/zenodo.12507609.

References

Agarwal, Raksha and Niladri Chatterjee. 2022. "Improvements in Multi-Document Abstractive Summarization Using Multi Sentence Compression with Word Graph and Node Alignment." *Expert Systems with Applications* 190 (March): 116154. doi: 10.1016/j.eswa.2021.116154.

Che, Wanxiang, Yanyan Zhao, Honglei Guo, Zhong Su, and Ting Liu. 2015. "Sentence Compression for Aspect-Based Sentiment Analysis." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23 (12): 2111–24. doi: 10.1109/TASLP.2015.2443982.

Chomsky, Noam 1957. *Syntactic Structures*. De Gruyter Mouton. doi: 10.1515/9783112316009.

Clarke, James and Mirella Lapata. 2008. "Global Inference for Sentence Compression: An Integer Linear Programming Approach." *Journal of Artificial Intelligence Research* 31 (March): 399–429. doi: 10.1613/jair.2433.

Cohn, Trevor and Mirella Lapata. 2008. "Sentence Compression Beyond Word Deletion." In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, edited by Donia Scott and Hans Uszkoreit, 137–44. Manchester, UK: Coling 2008 Organizing Committee. https://aclanthology.org/C08-1018.

Cohn, Trevor and Mirella Lapata. 2013. "An Abstractive Approach to Sentence Compression." *ACM Transactions on Intelligent Systems and Technology* 4 (3): 1–35. doi: 10.1145/2483669.2483674.

Corston-Oliver, Simon H. and William B. Dolan. 1999. "Less Is More: Eliminating Index Terms from Subordinate Clauses." In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 349–56. College Park, Maryland, USA: Association for Computational Linguistics. doi: 10.3115/1034678.1034734.

Crammer, Koby and Yoram Singer. 2001. "Ultraconservative Online Algorithms for Multiclass Problems." *In* Computational Learning Theory, edited by David Helmbold and Bob Williamson, Vol. 2111, 99–115. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-44581-1_7.

Galley, Michel and Kathleen McKeown. 2007. "Lexicalized Markov Grammars for Sentence Compression." In Human Language Technologies *2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, edited by Candace Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai, 180–87. Rochester, New York: Association for Computational Linguistics. https://aclanthology.org/N07-1023.

Gavhal, Trupti and Prachi Deshpande. 2023. "Sentence Compression Using Natural Language Processing." *SSRN Scholarly Paper* No. 4612736. Rochester, NY. doi: 10.2139/ssrn.4612736.

Ghayoomi, Masood. 2014. "From HPSG-Based Persian Treebanking to Parsing: Machine Learning for Data Annotation." doi: 10.17169/refubium-18160.

Grefenstette, Gregory. 1998. "Producing Intelligent Telegraphic Text Reduction to Provide an Audio Scanning Service for the Blind." In *Intelligent Text Summarization, AAAI Spring Symposium Series*, 111–17. https://cdn.aaai.org/Symposia/Spring/1998/SS-98-06/SS98-06-013.pdf.

Hasegawa, Shun, Yuta Kikuchi, Hiroya Takamura, and Manabu Okumura. 2017. "Japanese Sentence Compression with a Large Training Dataset." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, edited by Regina Barzilay and Min-Yen Kan, 281–86. Vancouver, Canada: Association for Computational Linguistics. doi: 10.18653/v1/P17-2044.

Hassel, Martin and Nima Mazdak. 2004. “FarsiSum - A Persian Text Summarizer.” *In Proceedings of the Workshop on Computational Approaches to Arabic Script-Based Languages*, 82–84. Geneva, Switzerland: COLING. https://aclanthology.org/W04-1615.

Honarpisheh, Mohamad Ali, Gholamreza Ghassem-Sani, and Ghassem Mirroshandel. 2008. “A Multi-Document Multi-Lingual Automatic Summarization System.” In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*. https://aclanthology.org/I08-2101.

Hou, Weiwei, Hanna Suominen, Piotr Koniusz, Sabrina Caldwell, and Tom Gedeon. 2020. “A Token-Wise CNN-Based Method for Sentence Compression.” In *Neural Information Processing*, edited by Haiqin Yang, Kitsuchart Pasupa, Andrew Chi-Sing Leung, James T. Kwok, Jonathan H. Chan, and Irwin King, 668–79. Cham: Springer International Publishing. doi: 10.1007/978-3-030-63830-6_56.

Jing, Hongyan. 2000. “Sentence Reduction for Automatic Text Summarization.” In *Sixth Applied Natural Language Processing Conference*, 310–15. Seattle, Washington, USA: Association for Computational Linguistics. doi: 10.3115/974147.974190.

Karimi, Zohre and Mehrnoush Shamsfard. 2006. “Summarization of Persian Texts.” *In Proceedings of 11th International CSI Computer Conference, Tehran, Iran*. http://faculty.du.ac.ir/z-karimi/pubs/758/.

Knight, Kevin and Daniel Marcu. 2002. “Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression.” *Artificial Intelligence* 139 (1): 91–107. doi: 10.1016/S0004-3702(02)00222-9.

Li, Zuchao, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2020. “Explicit Sentence Compression for Neural Machine Translation.” *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (5): 8311–18. doi: 10.1609/aaai.v34i05.6347.

Lin, Chin-Yew. 2004. “ROUGE: A Package for Automatic Evaluation of Summaries.” In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics. https://aclanthology.org/W04-1013.

Linke-Ellis, Nanci. 1999. “Closed Captioning in America: Looking beyond Compliance.” *In Proceedings of the TAO Workshop on TV Closed Captions for the Hearing Impaired People, Tokyo, Japan*, 43–59.

Masoumi, Saeid, Mohammad-Reza Feizi-derakhshi, and Raziyeh Tabatabaei. 2014. “Tabsum- A New Persian Text Summarizer.” *Journal of Mathematics and Computer Science* 11 (4): 330–42. doi: 10.22436/jmcs.011.04.08.

McDonald, Ryan. 2006. “Discriminative Sentence Compression with Soft Syntactic Evidence.” *In 11th Conference of the European Chapter of the Association for Computational Linguistics*, edited by Diana McCarthy and Shuly Wintner, 297–304. Trento, Italy: Association for Computational Linguistics. https://aclanthology.org/E06-1038.

Miao, Yishu and Phil Blunsom. 2016. “Language as a Latent Variable: Discrete Generative Models for Sentence Compression.” In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, edited by Jian Su, Kevin Duh, and Xavier Carreras, 319–28. Austin, Texas: Association for Computational Linguistics. doi: 10.18653/v1/D16-1031.

Napoles, Courtney, Benjamin Van Durme, and Chris Callison-Burch. 2011. “Evaluating Sentence Compression: Pitfalls and Suggested Remedies.” *In Proceedings of the Workshop on Monolingual Text-To-Text Generation*, edited by Katja Filippova and Stephen Wan, 91–97. Portland, Oregon: Association for Computational Linguistics. https://aclanthology.org/W11-1611.

Park, Yo-Han, Gyong-Ho Lee, Yong-Seok Choi, and Kong-Joo Lee. 2021. “Sentence Compression Using BERT and Graph Convolutional Networks.” *Applied Sciences* 11 (21): 9910. doi: 10.3390/app11219910.

Shamsfard, Mehrnoush, Tara Akhavan, and Mona Erfani Joorabchi. 2009. “Persian Document Summarization by Parsumist.” https://www.idosi.org/wasj/wasj7(c&it)/26.pdf.

Steinberger, Josef, and Karel Jezek. 2006. “Sentence Compression for the LSA-Based Summarizer.” *In Proceedings of the 7th International Conference on Information Systems Implementation and Modelling*, Vol. 180: 141–48. http://textmining.zcu.cz/publications/isim2006.pdf.

Tabatabayi Seifi, Shohreh and Iman Saraf Rezaee. 2017. “SAZEH: A Wide Coverage Persian Constituency Tree Bank and Parser.” In *Artificial Intelligence and Signal Processing (AISP)*.

Tan, Haochen, Han Wu, Wei Shao, Xinyun Zhang, Mingjie Zhan, Zhaohui Hou, Ding Liang, and Linqi Song. 2023. “Reconstruct Before Summarize: An Efficient Two-Step Framework for Condensing and Summarizing Meeting Transcripts.” *arXiv*. doi: 10.48550/arXiv.2305.07988.

Turner, Jenine and Eugene Charniak. 2005. “Supervised and Unsupervised Learning for Sentence Compression.” *In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, edited by Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, 290–97. Ann Arbor, Michigan: Association for Computational Linguistics. doi: 10.3115/1219840.1219876.

Yu, Naitong, Jie Zhang, Minlie Huang, and Xiaoyan Zhu. 2018. “An Operation Network for Abstractive Sentence Compression.” In *Proceedings of the 27th International Conference on Computational Linguistics*, edited by Emily M. Bender, Leon Derczynski, and Pierre Isabelle, 1065–76. Santa Fe, New Mexico, USA: Association for Computational Linguistics. https://aclanthology.org/C18-1091.

Zajic, David, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2007. “Multi-Candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks.” *Information Processing & Management*, Text Summarization, 43 (6): 1549–70. doi: 10.1016/j.ipm.2007.01.016.

Zajic, David M., Bonnie J. Dorr, and Jimmy Lin. 2008. “Single-Document and Multi-Document Summarization Techniques for Email Threads Using Sentence Compression.” *Information Processing & Management* 44 (4): 1600–1610. doi: 10.1016/j.ipm.2007.09.007.

Zhao, Yang, Zhiyuan Luo, and Akiko Aizawa. 2018. “A Language Model Based Evaluator for Sentence Compression.” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, edited by Iryna Gurevych and Yusuke Miyao, 170–75. Melbourne, Australia: Association for Computational Linguistics. doi: 10.18653/v1/P18-2028.