

With Measured Words: Simple Sentence Selection for Black-Box Optimization of Sentence Compression Algorithms

Yotam Shichel

Department of Software and
Information Systems Engineering
Ben Gurion University
yotamsh@post.bgu.ac.il

Meir Kalech

Department of Software and
Information Systems Engineering
Ben Gurion University
kalech@bgu.ac.il

Oren Tsur

Department of Software and
Information Systems Engineering
Ben Gurion University
orents@bgu.ac.il

Abstract

Sentence Compression is the task of generating a shorter, yet grammatical version of a given sentence, preserving the essence of the original sentence. This paper proposes a Black-Box Optimizer for Compression (B-BOC): given a black-box compression algorithm and assuming not all sentences need be compressed – find the best candidates for compression in order to maximize both compression rate and quality. Given a required compression ratio, we consider two scenarios: (i) single-sentence compression, and (ii) sentences-sequence compression. In the first scenario, our optimizer is trained to predict how well each sentence could be compressed while meeting the specified ratio requirement. In the latter, the desired compression ratio is applied to a sequence of sentences (e.g., a paragraph) as a whole, rather than on each individual sentence. To achieve that, we use B-BOC to assign an optimal compression ratio to each sentence, then cast it as a Knapsack problem, which we solve using bounded dynamic programming. We evaluate B-BOC on both scenarios on three datasets, demonstrating that our optimizer improves both accuracy and Rouge-F1-score compared to direct application of other compression algorithms.

1 Introduction

Sentence Compression is the task of generating a short, accurate, and fluent sentence that preserves the essence of a given original sentence by removing nonessential words and/or rephrasing it in a compact form. Compression can take many forms, ranging from Extractive and Abstractive Summarization (Jing, 2000; Madnani et al., 2007; Cohn and Lapata, 2008, 2009; Galanis and Androutsopoulos, 2010; Rush et al., 2015; Chopra

et al., 2016) to Text Simplification and Paraphrasing (Bannard and Callison-Burch, 2005; Xu et al., 2012; Klerke et al., 2016; Narayan et al., 2017; Aharoni and Goldberg, 2018; Botha et al., 2018), among others.

On the sentential level, compression is often viewed as a *word deletion* task (Knight and Marcu, 2000, 2002; Filippova and Strube, 2008; Filippova et al., 2015; Wang et al., 2016, 2017; Zhou and Rush, 2019). However, not all sentences could, or should be compressed as part of compressing a longer text they reside in. Consider the familiar scenario in which a full paragraph needs to be compressed in order to have an EACL paper meet the page restriction specified in the submission guidelines. A common approach by \LaTeX users is to first identify paragraphs ending with a short line, (e.g., this very paragraph), then choose one or more sentences that could be compressed with a minimal loss of information – shaving the extra line. We propose a Black-Box Optimizer for Compression (B-BOC) that mitigates this problem. Given a compression algorithm A , a desired compression ratio, and a document D , B-BOC chooses the best sentences to compress using A in order to produce a shorter version of D , while keeping the other sentences of D untouched. B-BOC achieves that without explicit knowledge of the inner-workings of the given compression algorithm, hence we call it a *black-box* optimizer. Selected sentences are expected to be the best candidates for compression – balancing compression rate with compression quality.

This paper addresses two main research questions: (1) How to predict the compression performance (preserving meaning and grammar) of an algorithm on a given sentence? (2) Given a document

and a required compression ratio, how to choose the optimal *subset* of sentences to compress, along with the appropriate compression ratio per each of the sentences, so that the *total* compression meets the required compression requirement?

Given a gold set of pairs of sentences and their compressions, we represent each sentence as a vector of shallow and syntactic features, and train a regression model to predict its expected compression rate. B-BOC ranks all sentences by the predicted compression potential while considering a required compression ratio.

The document-level task could be modeled as a Knapsack optimization problem, considering the subset of sentences to be compressed in order to satisfy the overall compression requirement (capacity), with a minimal loss of information (value). The solution space covers the trade-off between aggressively compressing only a few sentences and applying minimal compression on a larger number of sentences. While the general Knapsack is NP-complete, the 0-1 variation can be approximated efficiently by using Dynamic Programming (Hristakeva and Shrestha, 2005).

We evaluate B-BOC on three benchmarks commonly used for the sentence compression task. We show that applying B-BOC on top of state-of-the-art sentence compression models improves the performance for any desired compression rate. In addition, optimizing the B-BOC-Knapsack achieves top performance on the document-level task.

2 Related Work

Early sentence compression works employ the noisy channel model, learning the words and clauses to be pruned (Knight and Marcu, 2000, 2002; Filippova and Strube, 2008; Clarke and Lapata, 2008; Cohn and Lapata, 2009).

The top-performing sentence compression models use a Policy Network coupled with a Syntactic Language Model (bi-LSTM) evaluator (Zhou and Rush, 2019), and a stacked LSTM with dropout layers (Filippova et al., 2015). An extension of Filippova et al., adding syntactic features and using Integer Linear Programming (ILP), yields improved results in a cross-domain setting (Wang et al., 2017).

Sentence selection is used for document extractive summarization – a task conceptually close to ours, in which full sentences are extracted from a long document, see (Nenkova et al., 2011) for

an overview. State-of-the-art selection is achieved by combining sentence and document encoders (CNN and LSTM) with a sentence extraction model (LSTM) and a reinforcement layer (Narayan et al., 2018).

Sentence rephrasing is an abstractive approach to rewrite a sentence into a shorter form using some words that may not appear in the original sentence. A data-driven approach to abstractive sentence summarization is suggested in (Rush et al., 2015; Chopra et al., 2016), using about four million title-article pairs from the Gigaword corpus for training, and uses a convolutional neural network model to encode the source and produce a single representation for the entire input sentence. Tree-to-tree grammar extraction method for the rewriting task is used in (Cohn and Lapata, 2008, 2009). State-of-the-art performance on the abstractive summarization task is obtained using Hierarchical Attentional Seq2Seq Recurrent Neural Network (Nallapati et al., 2016; See et al., 2017).

3 Task Definitions and Methodology

In this section we formally define the sentence-level and the document-level tasks (§3.1) and provide a detailed description of the application of B-BOC in both settings (§3.2).

3.1 Problem Definitions

Sentence-Level Compression Given a set of sentences $S = \{s_i\}_{i=1}^n$; a desired compression rate γ ; the number of sentences to compress $k \leq n$; a compression algorithm A ; and an oracle $\mathbb{R} : (A, S) \rightarrow [0, 1]$, returning a score reflecting the compression quality (grammaticality and minimal loss of information) A would achieve on $s \in S$ – we would like to choose a set $S^{k,\gamma} \subseteq S$ of k sentences:

$$S^{k,\gamma} = \{s_j \mid \frac{|A(s_j)|}{|s_j|} \leq \gamma \wedge \operatorname{argmax}_{s_j} \mathbb{R}(A, s_j)\}_{j=1}^k.$$

We call this *sentence-level* compression since each sentence should meet the γ constraint independently. It is important to note that $\gamma' \leq \gamma \not\Rightarrow S^{k,\gamma} \subseteq S^{k,\gamma'}$, since different sentences may be better compressed to different γ values. Consider the following two sentences used to illustrate the importance of the Oxford comma: $S = \{“I had a yummy dinner with my parents, Batman and Catwoman”, “I had a yummy dinner with my parents, Batman, and Catwoman”\}^1$, and $k = 1$. The first

¹The first sentence, without the Oxford comma, implies that Batman and Catwoman are the speaker’s parents, the

sentence could be compressed to “I had a yummy dinner with my parents” with a minimal loss of information, while it does not make sense to compress the second sentence this way and it should be compressed to “I had a yummy dinner”, thus specifying $k = 1$, the sentence to be compressed with minimal loss of meaning depends on the desired γ value.

Document-Level Compression In this setting, we are given a sequence of sentences $D = \{s_i\}_{i=1}^n$ (a paragraph or a full document), and a desired compression rate γ that should be applied to D as a *whole*. That is, we wish to find an optimal subset of S^γ that satisfies:

$$\begin{aligned} \operatorname{argmax}_{S^\gamma \subseteq D} \sum_{s_i \in S^\gamma} \mathbb{R}(A, s_i) \\ \text{s.t.} \quad \frac{\sum_{s_i \in S^\gamma} |A(s_i)| + \sum_{s_i \in D \setminus S^\gamma} |s_i|}{|D|} \leq \gamma \end{aligned} \quad (1)$$

Since γ refers to D rather than to individual sentences, the overall quality can be maximized by choosing a varying number of sentences expected to achieve different optimal compressions. Unlike the sentence-level setting, here, an optimal S^γ may contain a combination of sentences, for some of which $\frac{|A(s)|}{|s|} \leq \gamma$, and for others $\frac{|A(s)|}{|s|} > \gamma$.

3.2 Computational Approach

Scoring Function Given a corpus $C = \{(s_i, \hat{s}_i)\}_i^m$ of sentence pairs, each pair contains an original sentence s and its gold compression \hat{s} , we define the golden ratio $\hat{\gamma}_i = \frac{|\hat{s}_i|}{|s_i|}$, and posit $\mathbb{R}(A, s) \approx 1 - |\hat{\gamma}_i - \frac{|A(s_i)|}{|s_i|}|$.

We justify the use of $\hat{\gamma}_i$ as a proxy to the optimal compression quality, as compression ratios are found to correlate with compression quality measured against gold compressions (Napoles et al., 2011). The use of $\hat{\gamma}_i$ as a proxy is validated through manual evaluation, see Sections 4.2 and 5.1.

Syntactic features were successfully used for sentence compression (Clarke and Lapata, 2008; Wang et al., 2017; Liu et al., 2017; Futrell and Levy, 2017). Assuming that sentence complexity correlates with the ease of compression, we follow (Brunato et al., 2018) and represent each sentence as a vector of shallow features (sentence length, average word length, punctuation counts, etc.) and

second sentence implies that the speaker had dinner with four people – her parents *and* Batman *and* Catwoman.

syntactic features (depth of constituent parse tree as well as the number of internal nodes, word’s depth in a dependency parse tree, mean dependency distance, etc.).

We now train a regression model and learn the scoring function $R(A, s)$ by minimizing the loss:

$$L(C, A) = \sum_{s_i \in C} [R(A, s_i) - \mathbb{R}(A, s_i)]^2$$

We note that we do not train a compression algorithm, but an oracle – a scoring function that predicts the *quality* of the compression algorithm A will achieve on a given sentence. This oracle will be used to rank candidate sentences in order to optimize the choice of sentences in the two tasks defined in Section 3.1.

We train a Gradient Boosted Tree regression model using XGBoost. The model’s hyperparameters (e.g., subsample ratio, learning rate, max depth) were tuned on a separated development set.

Sentence level compression: Given a set of sentences S , B-BOC operates on two steps: (i) It applies R on every $s \in S$, producing an ordered set \hat{S} for which $\forall_{i < j} R(A, s_i) \geq R(A, s_j)$. (ii) It constructs $S^{k, \gamma}$ by iterating over \hat{S} , choosing the first k sentences that satisfy the γ requirement.

Document level compression: Using the task definition in Section 3.1, it is straight forward to cast the task as a combinatorial 0-1 Knapsack problem in the following way: Given a set of items (sentences) $S = \{s_1, \dots, s_n\}$, each weighs $w_i = |A(s_i)|$ if compressed, or $|s_i|$ if kept in the original form, and each holds a value $v_i = R(A, s_i)$ (predicted compression quality), if compressed and $v_i = 1$ if kept in the original form; and given a weight limit $W = \gamma \cdot \sum_1^n |s_i|$ – we wish to find $S^\gamma = \{s_i | x_i = 1\}$ that maximizes:

$$\begin{aligned} \sum_i (v_i x_i - [1 - x_i]^{-1}) \quad \text{s.t.} \\ \sum_i (w_i x_i - |s_i| [1 - x_i]^{-1}) \leq W \quad , x_i \in \{0, 1\} \end{aligned}$$

where $x_i = 1$ denotes we choose to compress s_i and $x_i = 0$ denotes that s_i remains in its original long form (hence the 0-1 Knapsack setting). Note that the value we maximize and the weight constraints include a term for the unchanged sentences, in case they are not chosen for compression. This term is introduced since the γ constraint in the task definition applies to the document as a whole.

B-BOC-knapsack returns S^γ by solving the 0-1 knapsack problem using the dynamic programming approach proposed by (Hristakeva and Shrestha, 2005) to reduce the computation complexity to a pseudo-polynomial time. Knapsack’s solution ensures an optimal set of sentences, satisfying the required compression limitations, while achieving the maximum quality score.

4 Experimental Setting

4.1 Datasets

Training Data: We train B-BOC on a dataset of 200,000 sentence-compression pairs² used by Filippova and Altun (2013). Each pair is composed of a long sentence (usually the teaser, caption, extract or the first sentence that bears the most salient information) from a news story and the story’s headline, which is a compressed version of the long sentence.

Out of these 200,000 sentences, we set aside 9,000 to be used as a development set, and 1,000 as one of our three test sets.

Evaluation datasets: Three datasets are used for evaluation:

1. Google (GGL) – the first 1000 sentences of the training corpus (described above) were used for testing.
2. British National Corpus (BNC) – a manually crafted dataset of ~ 1500 sentence-compression pairs. Given a long sentence, annotators were asked to produce a short version by deleting extraneous words from the source without changing the order of words³.
3. Gigaword (GIGA)- headline-generation corpus of articles⁴ consists ~ 4 million sentence-compression pairs. We note that this dataset contains abstractive pairs, nevertheless, it can be used to measure accuracy.

4.2 Evaluation Procedures

Evaluation metrics: We used four evaluation metrics that complement each other, providing a comprehensive evaluation of the different factors that contribute to quality summarization as suggested by (Filippova et al., 2015): (1) Accuracy – how many compressed sentences are fully reproduced, (i.e., the generated compression is identical

²www.github.com/google-research-datasets/sentence-compression

³jamesclarke.net/research/resources

⁴github.com/harvardnlp/sent-summary

to the golden one). (2) F-score – given the golden and predicted compressions, recall and precision are based on the ROUGE metric. (3) Readability score – the grammaticality of the compression. (4) Informativeness – the level in which the compression covers the most salient information.

The two latter metrics are based on a manual evaluation by three annotators, scoring Readability and Informativeness on a 5-Point Likert scale. The annotators were guided to give a top Readability score (score 5) if the predicted compressed sentence is clear and grammatically correct, regardless of the original context, and a top Informativeness score (score 5) if the essence of the original is preserved completely. The Informativeness measurement bears some degree of subjectivity as annotators may not agree on what should be considered “the essence” of a sentence, see examples in Table 1. We used Cohen’s Kappa (Cohen, 1960) to measure inter-annotator agreements. Low agreements are expected due to the subjectivity and the five-point scale, i.e., when two raters agree on the grammaticality of a sentence, but do not give the same exact Informativeness score. To account for slight variations in assessment, we measure agreement using the off-by-one procedure proposed by (Tsur and Rappoport, 2009) and supported by (Toutanova et al., 2016). Linear and Quadratic weighting were added as additional statistical methods. Nevertheless, we kept the 5-point scale to be aligned with Filippova’s evaluations. The Kappa values for the strict and the off-by-one agreement for a sample of 200 sentences of the GGL dataset are reported in Table 2. These scores are comparable with the scores reported by (Filippova et al., 2015). Agreement of 0.86 and 0.78 for Readability and Informativeness reflect an almost perfect agreement on Readability and substantial agreement on Informativeness, according to the interpretation protocol suggested by McHugh (2012).

4.2.1 Black-Box Compression Models

As described in Section 3.2, B-BOC accommodates any compression model used to compress the sentences. To show this independence, B-BOC is evaluated with three competitive compression models: (1) Filippova: An LSTM model trained on two million sentence-compression pairs (Filippova et al., 2015), (2) Zhou: An unsupervised model for sentence summarization (Zhou and Rush, 2019), and (3) Klerke: A three-layer bi-LSTM model (Klerke et al., 2016).

Source	Text	Issue
Long	A gang of youths between eight and sixteen robbed a man in an Oldbrook underpass for just 10£	The salience the clause “for just 10£”
Manual 1	<i>A gang of youths robbed a man in an Oldbrook underpass for just 10£</i>	
Manual 2	<i>A gang robbed a man in an Oldbrook underpass</i>	
Long	A woman was injured by a falling tree in the Gresham neighborhood, according to the Chicago Fire Department	The salience of the location “Gresham neighborhood”
Manual 1	<i>A woman was injured by a falling tree</i>	
Manual 2	<i>A woman was injured by a falling tree in the Gresham neighborhood</i>	

Table 1: Two examples of compression disagreements.

Agreement coverage	Readability	Informativeness
Strict	0.61	0.32
Off-by-one	0.86	0.78
Linear	0.78	0.54
Quadratic	0.87	0.72

Table 2: Cohen’s Kappa inter-annotator agreement between three annotators for the strict, off-by-one, and other statistical approaches to calculate agreement.

4.3 Experimental Procedure

Given the two settings presented in Section 3.1, we aim to evaluate the performance of B-BOC in optimizing compression quality, on top of a number of black-box compression algorithms. We evaluate the way different values of k affect the performance, and explore the contribution of various feature types to the trained optimizer.

4.3.1 Sentence level compression:

We evaluate the effectiveness of B-BOC for varied compression rates. The tested sentences were divided into buckets of different compression rates 0.1-0.9. For each bucket we set k to be 50% of the sentence in a bucket and compare B-BOC selections to: (1) A random selection of k sentences from the bucket (RANDOM). (2) The average of all sentences in the compression rate bucket (ALL). We report results of this comparison for each of the black-box algorithms listed in Section 4.2.1). Note that the F-score is based on the actual results of each of the black-box models, and that both B-BOC and RANDOM choose from the same pool of candidates for each compression rate bucket.

4.3.2 Document level compression

Having a document or a paragraph comprised of several sentences that are needed to be compressed, the target is to find the sentences that would gain the highest performance score subject to the overall compression ratio constraint.

To simulate a document, we synthesized one hundred documents by randomly sampling sentences from the test set. Every document contains 100 different sentences of varying lengths. We then use B-BOC-Knapsack as described in 3.2. B-BOC-Knapsack is compared with: (1) an oracle Knapsack solution where the golden scores are provided, rather than estimated by B-BOC (ORACLE). (2) We iteratively sample sentences to compress until the compression ratio is reached (RANDOM). (3) A sorted selection- choosing sentences by their lengths in an ascending sort (SHORTER FIRST). The latter baseline was added followed by our experiments, showing that compression quality tends to be higher for shorter sentences.

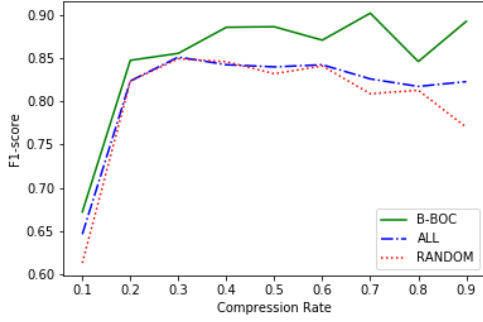
5 Results and Discussion

5.1 Results

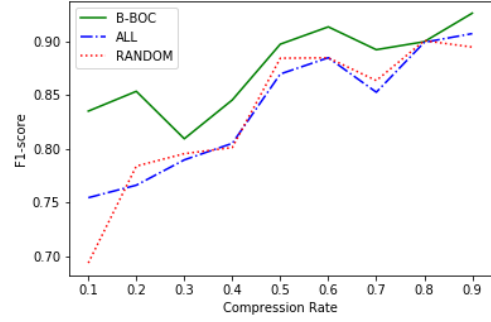
Detailed results for both sentence level and document level compression are presented below.

Sentence level compression: Figure 1 presents the F1 performance of sentence selection methods over varied γ -buckets on the GGL dataset, while training with the compression models of Filippova and Zhou respectively. B-BOC is compared with all sentences and a random selection of sentences, as described in Section 4.3.1. It can be seen that B-BOC achieves the highest F1-score for every γ . The evaluated metrics’ averages for all compression buckets are presented in Table 3, evaluating the GGL dataset using three different compression models. Best results are in bold. B-BOC achieves the best performance overall measures – automatic and manual (F1-score, Accuracy, Readability and Informativeness).

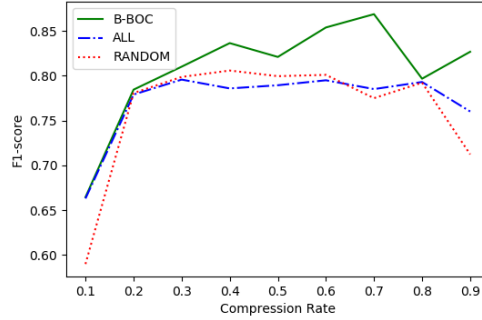
The results confirm that by utilizing B-BOC, the top sentences which yield the best overall compression results will be chosen, no matter which black-box compression model is applied, for every given compression ratio. Table 4 describes the average F1-scores and variances for the manual evaluations



(a) Filippova's compression model



(b) Zhou's compression model



(c) Klerke's compression model

Figure 1: Average F1-score (y -axis) applied on the GGL dataset for different compression rate buckets (x -axis) while training B-BOC with Filippova (a), Zhou's (b) and Klerke (c) compression models.

		F1-score	F2-score	Accuracy	Readability	Info.
Filippova	ALL	0.837	0.77	0.31	4.562	3.78
	RANDOM	0.835	0.76	0.306	4.559	3.79
	B-BOC	0.86 *	0.795	0.332	4.65	4.08
Zhou	ALL	0.82	0.7	0.24	3.92	3.41
	RANDOM	0.815	0.69	0.226	3.91	3.41
	B-BOC	0.87 *	0.77	0.30	3.92	3.60
Klerke	ALL	0.787	0.685	0.187	4.12	3.73
	RANDOM	0.783	0.677	0.156	4.00	3.65
	B-BOC	0.815 *	0.72	0.214	4.23	3.97

Table 3: GGL dataset: Evaluation metrics' average results over all compression rate buckets. Statistical significance using a paired T-test is indicated by *.

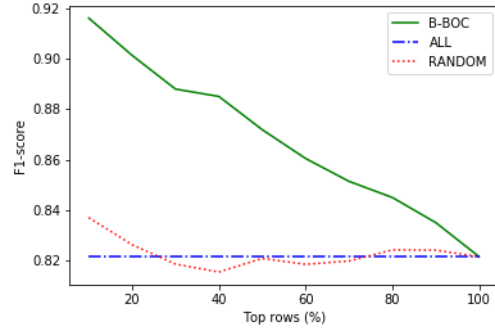


Figure 2: F1-score (y -axis) per top $X\%$ of sentences (x -axis) that are ranked by B-BOC.

Likert score	Info. F1	Read. F1	Info. Var	Read. Var
1	0.59	0.67	0.13	0.10
2	0.66	0.66	0.07	0.04
3	0.70	0.69	0.05	0.05
4	0.77	0.74	0.04	0.05
5	0.77	0.75	0.03	0.04

Table 4: Readability and Informativeness average F1-scores and variance.

of the GGL dataset using Filippova's compression model. It can be seen that both Readability and Informativeness are correlated with F1-scores. A compression that gets a higher Readability or Informativeness on the 5-point Likert score, will most probably get a higher F1-score as well, with a lower variance. This manual evaluation of Readability and Informativeness supports our choice of \mathbb{R} (see Section 3.2).

As described in Section 4.3.1, the top 50% of the test dataset are chosen for our evaluations. We

repeated the same experiment varying the number (percentage) of sentences ranked by B-BOC. Figure 2 presents the impact of the number of sentences we consider, and demonstrates the deterministic trend of B-BOC ranking method. It can be seen that when considering only the higher ranked sentences, their compression will produce a higher F1-score. It suggests that the lower number of sentences we consider – the higher the benefit of B-BOC is, compared with a random selection of the same number of selected sentences.

Document level compression: Given a document or a paragraph and a specified compression rate requirement, B-BOC-Knapsack aims to find a subset of sentences, that together will satisfy the compression rate constraints if being compressed, and while guaranteeing a top F1-score. Our results below depict an experiment for compressing a document with a certain compression ratio constraint. A document is constructed using 100 sentences with variate lengths, randomly selected from a given dataset. B-BOC-Knapsack sentence selection is being compared with a random selection and a sorted selection of the sentences, as described in Section 4.3.2. Each experiment was repeated 100 times, sampling different sentences for each of the datasets. The average scores reported below were achieved with the same compression model used by Filippova (see Section 4.2.1) for all sentences.

Figure 3 presents the experiments for GGL and BNC datasets respectively. An overall compression requirement is added, ranging from 0.1 to 0.5 (e.g., 0.1 means that the document should be compressed in 10 percent). B-BOC-Knapsack has a higher F1-score for almost every compression ratio, especially at the lower ratios.

Knapsack’s oracle solution can be created when considering the actual F1 and compression rates for all sentences. A histogram of the sentences that the oracle Knapsack chose to compress, grouped by their lengths is presented in Figure 4. The Figure provides a number of insights: (1) The F1-score decreases as the number of compressed sentences grow, due to the increased uncertainty when compressing more sentences. A similar pattern is observed in Figure 3. (2) The Knapsack prefers to choose shorter sentences, as these perform better than longer sentences. We attribute this to the fact that shorter sentences may be easier to optimize, as compression alternatives are limited, compared to longer sentences.

		F1-score	Readability	Informativeness
GGL	ORACLE	0.89	4.64	4.05
	RANDOM	0.837	4.59	3.76
	SHORTER FIRST	0.850	4.6	3.97
	B-BOC	0.854	4.62	4.01
BNC	ORACLE	0.68	3.77	3.26
	RANDOM	0.55	3.53	3.13
	SHORTER FIRST	0.62	3.69	3.24
	B-BOC	0.63	3.79	3.28
GIGA	ORACLE	0.47	-	-
	RANDOM	0.256	-	-
	SHORTER FIRST	0.297	-	-
	B-BOC	0.303	-	-

Table 5: Average results- Document level compression.

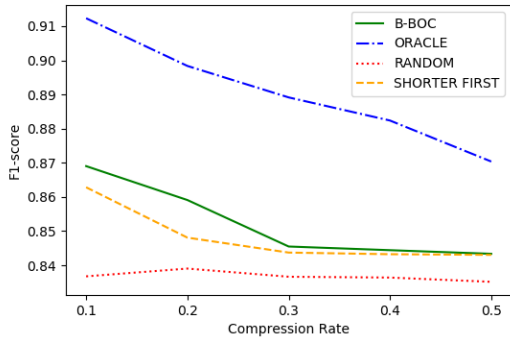
The average results for the three datasets are presented in Table 5. Best results are in bold. Informativeness and Readability average scores are aligned with the F1-scores (note that the GIGA dataset was not manually annotated for Readability and Informativeness, since we are focused on extractive summarization rather than abstractive, and the Readability and Informativeness of the two types cannot be compared directly). We observe that B-BOC chooses the best sentences and provides a better compression performance for any compression ratio.

Feature Importance: Sentence complexity is correlated with the parse tree structure (Oya, 2011). Analyzing the contribution of each feature type, we find the tree depth features and especially Mean Dependency Distance (MDD) to do the heavy lifting. The MDD is the sum of the depth of words in the dependency tree, divided by the total number of dependencies. For example, the MDD scores for two sentences of the same character length “Sarah read the book quickly and understood it correctly” (Figure 5 top) and “US President Donald Trump tests positive for coronavirus” (Figure 5 bottom) is $19/8 = 2.735$ and $11/7 = 1.57$, respectively. This observation validates the relation between sentence complexity and compression.

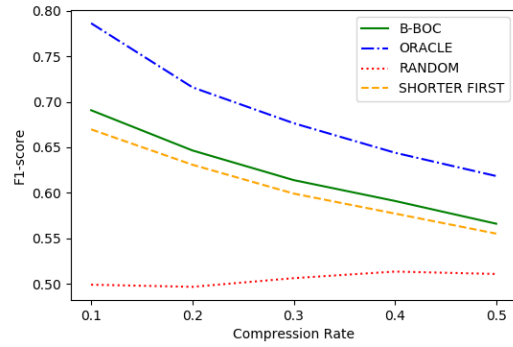
Table 6 presents the importance of the syntactic features to the B-BOC model in terms of weight, which means the relative number of times a feature occurs in the boosted trees of the trained model. Shallow properties such as the number of verbs and number of nodes are located at the bottom.

5.2 Discussion

Limitation of the F1-score Our main target is maximizing the F1-score, which happens to be a common approach for the sentence compression task, e.g., (Filippova et al., 2015; Zhao et al., 2018).



(a) GGL dataset



(b) BNC dataset

Figure 3: 0-1 Knapsack’s F-score results for GGL dataset (a) and BNC dataset (b). x -axis is the total desired compression rate of the document (i.e., 0.1 means compressing the whole document by 10 percent). y -axis is the average F1-score of the subset of sentences being compressed.

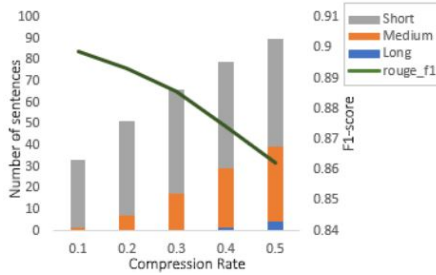


Figure 4: GGL dataset: Oracle Knapsack subset’s histogram. Two y -axis are average F1-score of the subset (right axis, describes the line) and the number of selected sentences to be compressed (left axis, describes the bars’ height). x -axis is the total desired compression rate of the document.

Feature name	Weight
Word Length Average	0.140630
MDD-score	0.121428
Tree Depth Average	0.116776
Character count	0.093331
Count Relations	0.079562
Count PoS	0.072715
Parse-Tree Height	0.069478
Count Nodes	0.066835
Parse-Tree sub trees	0.061700
Count words	0.051541
Dependency tree depth	0.045512
Verb count	0.042461
Parse-Tree count POS types	0.038032

Table 6: Feature Importance of B-BOC. The percentages representing the relative number of times a particular feature occurs in the trees of the model.

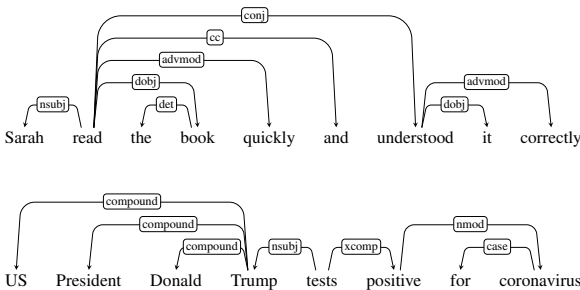


Figure 5: Dependency trees of sentences of the same length (chars), but different depth and MDD score.

Automatic evaluation metrics like the F1-score serve complementary purposes for linguistic quality evaluation rather than replacement because it is unclear whether the improvement in F1-score necessarily indicates the improvement of linguistic quality. Nevertheless, it was shown that the F1-score correlates with human judgment (Napoles et al., 2011). We manually performed additional evaluation for Readability and Informativeness to complement the evaluation based on the F1-score. For example, when applied on the document-level on the BNC dataset, B-BOC does not achieve the best F-score but does achieve best Readability and Informativeness scores (see Table 5).

Fairness Compression algorithms should be compared for similar levels of compression (Napoles et al., 2011). Partitioning S to different compression rate buckets, as explained in 4.3.1 and

demonstrated in Figure 1, ensures a fair comparison between the different compression models.

Manual evaluations. Exploring the cases in which annotators did not agree on either Readability or Informativeness, we noticed a higher likelihood for disagreement in the lower scale of both measurements, especially in case the original sentence was convoluted or grammatically flawed.

6 Conclusions

In this paper we presented B-BOC- Black-Box Optimizer for Compression, a new complexity optimization method designated to the sentence compression problem. We defined the correlation between the complexity of a sentence and the chance that a black-box compression model could successfully compress it. Our optimization model is independent of the compression model used to compress the sentences and can be combined with any sentence compression model. Our evaluation on three benchmarks revealed promising results when applied to three different types of sentence compression models. We achieve top performance for a document compression problem using the B-BOC-Knapsack optimization implemented with a bounded Dynamic Programming technique. Our method could assist in compressing any kind of text while applying their desired compression model. Utilizing our method provides a proper guideline for which of the sentences are the most beneficial to focus on, in order to compress a given text, while yielding the best overall compression results.

For future work, we plan to construct Model-Dependant Optimization that accounts for the features of each compression model. This will facilitate a choice of the compression model that is the most suitable for a given sentence.

Acknowledgments

We would like to thank Katja Filippova and Alexander Rush for providing us with the outputs of their compression models.

References

Roei Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for*

Computational Linguistics, pages 597–604. Association for Computational Linguistics.

Jan A Botha, Manaal Faruqui, John Alex, Jason Baldrige, and Dipanjan Das. 2018. Learning to split and rephrase from wikipedia edit history. *arXiv preprint arXiv:1808.09468*.

Dominique Brunato, Lorenzo De Mattei, Felice Dell’Orletta, Benedetta Iavarone, and Giulia Venturi. 2018. Is this sentence difficult? do you agree? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2690–2699.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics.

Trevor Anthony Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368.

Katja Filippova and Yasemin Altun. 2013. [Overcoming the lack of parallel data in sentence compression](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, Seattle, Washington, USA. Association for Computational Linguistics.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32. Association for Computational Linguistics.

Richard Futrell and Roger Levy. 2017. [Noisy-context surprisal as a human sentence processing cost model](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 688–698,

- Valencia, Spain. Association for Computational Linguistics.
- Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 885–893. Association for Computational Linguistics.
- Maya Hristakeva and Dipti Shrestha. 2005. Different approaches to solve the 0/1 knapsack problem. In *The Midwest Instruction and Computing Symposium*.
- Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Sixth Applied Natural Language Processing Conference*.
- Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1528–1533, San Diego, California. Association for Computational Linguistics.
- Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization-step one: Sentence compression. *AAAI/IAAI*, 2000:703–710.
- Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.
- Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews*, 21:171–193.
- Nitin Madnani, David Zajic, Bonnie Dorr, Necip Fazil Ayan, and Jimmy Lin. 2007. Multiple alternative sentence compressions for automatic text summarization. In *Proceedings of DUC*.
- Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282.
- Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.
- Courtney Napoles, 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*, pages 91–97.
- Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv preprint arXiv:1802.08636*.
- Shashi Narayan, Claire Gardent, Shay B Cohen, and Anastasia Shimorina. 2017. Split and rephrase. *arXiv preprint arXiv:1707.06971*.
- Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.
- Masanori Oya. 2011. Syntactic dependency distance as sentence complexity measure. In *Proceedings of the 16th International Conference of Pan-Pacific Association of Applied Linguistics*, pages 313–316.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.
- Kristina Toutanova, Chris Brockett, Ke M. Tran, and Saleema Amershi. 2016. A dataset and evaluation metrics for abstractive compression of sentences and short paragraphs. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 340–350, Austin, Texas. Association for Computational Linguistics.
- Oren Tsur and Ari Rappoport. 2009. Revrnk: A fully unsupervised algorithm for selecting the most helpful book reviews. In *ICWSM*. Citeseer.
- Liangguo Wang, Jing Jiang, Hai Leong Chieu, Chen Hui Ong, Dandan Song, and Lejian Liao. 2017. Can syntax help? improving an lstm-based sentence compression model for new domains. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1385–1393.
- Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2016. A sentence compression based framework to query-focused multi-document summarization. *arXiv preprint arXiv:1606.07548*.
- Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. *Proceedings of COLING 2012*, pages 2899–2914.
- Yang Zhao, 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)*, pages 170–175.
- Jiawei Zhou and Alexander Rush. 2019. Simple unsupervised summarization by contextual matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy. Association for Computational Linguistics.