Multi-Sentence Compression with Word Vertex-Labeled Graphs and Integer Linear Programming

Elvys Linhares Pontes[1], **Stéphane Huet**[1], **Thiago Gouveia da Silva**[1,3,4]
and **Juan-Manuel Torres-Moreno**[1,2]
1 - LIA, Université d'Avignon et des Pays de Vaucluse, Avignon, France
2 - École Polytechnique de Montréal, Montréal, Canada
3 - Inst. de Computação – Univ. Federal Fluminense (UFF), Niterói – RJ – Brazil
4 - Inst. Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), PB – Brazil
elvys.linhares-pontes@alumni.univ-avignon.fr
stephane.huet@univ-avignon.fr
thiago.gouveia@ifpb.edu.br
juan-manuel.torres@univ-avignon.fr

Andréa C. Linhares
Universidade Federal do Ceará, Sobral, Brazil
andrea@sobral.ufc.br

Abstract

Multi-Sentence Compression (MSC) aims to generate a short sentence with key information from a cluster of closely related sentences. MSC enables summarization and question-answering systems to generate outputs combining fully formed sentences from one or several documents. This paper describes a new Integer Linear Programming method for MSC using a vertex-labeled graph to select different keywords, and novel 3-grams scores to generate more informative sentences while maintaining their grammaticality. Our system is of good quality and outperforms the state-of-the-art for evaluations led on news dataset. We led both automatic and manual evaluations to determine the informativeness and the grammaticality of compressions for each dataset. Additional tests, which take advantage of the fact that the length of compressions can be modulated, still improve ROUGE scores with shorter output sentences.

1 Introduction

The increased number of electronic devices (smartphones, tablets, etc.) have made access to information easier and faster. Websites such as Wikipedia or news aggregators can provide detailed data on various issues but texts may be long and convey a lot of information. One solution to this problem is the generation of summaries containing only key information.

Among the various applications of Natural Language Processing (NLP), Automatic Text Summarization (ATS) aims to automatically identify the relevant data inside one or more documents, and create a condensed text with the main information. Summarization systems usually rely on statistical, morphological and syntactic analysis approaches (Torres-Moreno, 2014). Some of them use Multi-Sentence Compression (MSC) in order to produce from a set of similar sentences a small-sized sentence which is both grammatically correct and informative (Filippova, 2010; Banerjee et al., 2015). Although compression is a challenging task, it is appropriate to generate summaries that are more informative than state-of-the-art extraction methods for ATS.

The contributions of this article are two-fold. (i) We present a new model for MSC that extends the common approach based on Graph Theory, using vertex-labeled graphs and Integer Linear Programming (ILP) to select the best compression. The vertex-labeled graphs are used to model a cluster of similar sentences with keywords, while the optimization criterion introduces a novel 3-grams score to enhance the correctness of sentences. (ii) We can set up a maximum length for the compression. The system can generate shorter compressions losing some information, or privilege the informativeness generating longer compressions. Evaluations led with both automatic metrics and human evaluations show that our ILP model consistently generate more informative sentences than two baselines while maintaining their grammaticality. Our approach is able to choose the amount of information to keep in the compression output, through the definition of the number of keywords

Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12), pages 18–27
New Orleans, Louisiana, June 6, 2018. ©2018 Association for Computational Linguistics

to consider in documents.

This paper is organized as follows: we describe and survey the MSC problem in Section 2. Next, we detail our approach in Section 3. The experiments and the results are discussed in Sections 4 and 5. Lastly, we provide the Conclusion and some final comments in Section 6.

2 Related Work

Sentence Compression (SC) aims at producing a reduced grammatically correct sentence. Compressions may have different Compression Ratio (CR) levels[1], whereby the lower the CR level, the higher the reduction of the information is. SC can be employed in the contexts of the summarization of documents, the generation of article titles or the simplification of complex sentences, using diverse methods such as optimization (Clarke and Lapata, 2007, 2008), syntactic analysis, deletion of words (Filippova et al., 2015) or generation of sentences (Rush et al., 2015; Miao and Blunsom, 2016).

Multi-Sentence Compression (MSC), also coined as Multi-Sentence Fusion, is a variation of SC. Unlike SC, MSC combines the information of a cluster of similar sentences to generate a new sentence, hopefully grammatically correct, which compresses the most relevant data of this cluster. The idea of MSC was introduced by Barzilay and McKeown (2005), who developed a multi-document summarizer which represents each sentence as a dependency tree; their approach aligns and combines these trees to fusion sentences. Filippova and Strube (2008) also used dependency trees to align each cluster of related sentences and generated a new tree this time with ILP to compress the information. In 2010, Filippova presented a new model for MSC, simple but effective, which is based on Graph Theory and a list of stopwords. She used a Word Graph (WG) to represent and to compress the related sentences of the document D based on the cohesion of words. The vertices and the arcs weights of WG represent the word/POS pairs and the levels of cohesion between two words in the document (Equation 1), respectively.

$$w(i,j) = \frac{\text{cohesion}(i,j)}{\text{freq}(i) \times \text{freq}(j)}, \tag{1}$$

$$\text{cohesion}(i,j) = \frac{\text{freq}(i) + \text{freq}(j)}{\sum_{s \in D} \text{diff}(s,i,j)^{-1}}, \tag{2}$$

where freq(i) is the word frequency mapped to the vertex i and the function diff(s,i,j) refers to the distance between the offset positions of words i and j in the sentences s of D containing these two words. Finally, she chose as the best MSC the path with the lowest average arc weight among the 50 shortest paths (more details in (Filippova, 2010)).

Inspired by the good results of the Filippova's method, many studies have used it in a first step to generate a list of the N shortest paths, then have relied on different reranking strategies to analyze the candidates and select the best compression (Boudin and Morin, 2013; Tzouridis et al., 2014; Luong et al., 2015; Banerjee et al., 2015). Boudin and Morin (2013) developed a reranking method measuring the relevance of a candidate compression using *keyphrases*, obtained with the TextRank algorithm (Mihalcea and Tarau, 2004), and the length of the sentence. Another reranking strategy was proposed by Luong et al. (2015). Their method ranks the sentences from the counts of unigrams occurring in every source sentence. ShafieiBavani et al. (2016) also used a WG model; their approach consists of three main components: (i) a merging stage based on Multiword Expressions (MWE), (ii) a mapping strategy based on synonymy between words and (iii) a reranking step to identify the best compression candidates generated using a POS-based language model (POS-LM). Tzouridis et al. (2014) proposed a structured learning-based approach. Instead of applying heuristics as Filippova (2010), they adapted the decoding process to the data by parameterizing a shortest path algorithm. They devised a structural support vector machine to learn the shortest path in possibly high dimensional joint feature spaces and proposed a generalized, loss-augmented decoding algorithm that is solved exactly by ILP in polynomial time.

We found two other studies that applied ILP to combine and compress several sentences. Banerjee et al. (2015) developed a multi-document ATS system that generated summaries based on compression of similar sentences. They used Filippova's method to generate 200 random compressed sentences. Then they created an ILP model to select the most informative and grammatically correct compression. Thadani and McK-

[1] The CR is the length of the compression divided by the average length of all source sentences

eown (2013) proposed another ILP model using an inference approach for sentence fusion. Their ILP formulation relies on n-grams factorization and aims at avoiding cycles and disconnected structures.

Following previous studies that rely on Graph Theory with good results, this work presents a new ILP framework that takes into account keywords and 3-grams for MSC. We compare our learning approach to the graph-based sentence compression techniques proposed by Filippova (2010) and Boudin and Morin (2013), considered as state-of-the-art methods for MSC. We intend to apply our method on various languages and not to be dependent on linguistic resources or tools specific to languages. This leads us to put aside systems which, despite being competitive, rely on resources like WordNet or Multiword expression detectors (ShafieiBavani et al., 2016).

3 Our Approach

Filippova's method chooses the path in a WG with the lowest score taking into account the level of cohesion between two adjacent words in the document. However, two words with a strong cohesion do not necessarily have a good informativeness because the cohesion only measures the distance and the frequency of words in the sentences. In this work, we propose a method to concurrently analyze cohesion, keywords and 3-grams in order to generate a more informative and comprehensible compression.

Our method calculates the shortest path from the cohesion of words and grants bonuses to the paths that have different keywords and frequent 3-grams. For this purpose, our approach is based on Filippova's method to model a document D as a graph and to calculate the cohesion of words. In addition, we analyze the keywords and the *3-grams* of the document to favor hypotheses with meaningful information.

3.1 Keyword and 3-grams extraction

Introducing keywords in the graph helps the system to generate more informative compressions because it takes into account the words that are representative of the cluster to calculate the best path in the graph, and not only the cohesion and frequency of words. Keywords can be identified for each cluster with various extraction methods and we study three widely used techniques: Latent Semantic Indexing (LSI), Latent Dirichlet Allocation (LDA) and TextRank. Despite the small number of sentences per cluster, these methods generate good results because clusters are composed of similar sentences with a high level of redundancy. LSI uses Singular-Value Decomposition (SVD), a technique closely related to eigenvector decomposition and factor analysis, to model the associative relationships (Deerwester et al., 1990). LDA is a topic model that generates topics based on word frequency from a set of documents (Blei et al., 2003). Finally, TextRank algorithm analyzes the words in texts using WGs and estimates their relevance (Mihalcea and Tarau, 2004). For LDA whose modeling is based on the concept of topics, we consider that the document D describes only one topic since it is composed of semantically close sentences. A same word or keyword can be represented by one or several nodes in WGs (see the WG construction in (Filippova, 2010)). In order to prioritize the sentence generation containing keywords, we add a bonus to the compression score when the compression contains different keywords. This process favors informativeness but may neglect grammaticality. Therefore, we also analyze 3-grams and compute in the graph their relevance scores.

The presence of a word in different sentences is assumed to increase its relevance for the MSC (we do not analyze stopwords). Thus, we define the relevance of a word i according to Equation 3.

$$\text{1-grams}(i) = \frac{\text{freq}(i)}{|D|_w} \times \frac{\text{freq}_s(i)}{|D|_s} \tag{3}$$

where $\text{freq}_s(i)$ is the number of sentences containing the word i, $|D|_w$ and $|D|_s$ are the overall number of word occurrences and the number of sentences in the document D, respectively. Since we analyze Word Graphs whose basic connections are arcs associated with 2-grams, we define the relevance of 3-grams[2] from the inner 2-grams (Eq. 4 and 5).

$$\begin{aligned}\text{3-grams}(i,j,k) = \text{freq}_3(i,j,k) \times \\ \frac{\text{2-grams}(i,j) + \text{2-grams}(j,k)}{2}\end{aligned} \tag{4}$$

[2] Since clusters are small, they have a limited number of sequences of an order higher than 3. Therefore, the use of 4-grams increases the complexity of the model without improving the quality of compression.

$$\text{2-grams}(i,j) = \frac{\text{1-grams}(i) + \text{1-grams}(j)}{2} \quad (5)$$

where $\text{freq}_3(i,j,k)$ is the amount of 3-grams composed of the words i, j and k in the document. Taking into account frequent 3-grams aims at improving the grammatical quality of MSC while keeping the relevant information. The 3-grams bonus is obtained from the relevance of the 3-grams (Eq. 4).

3.2 Vertex-Labeled Graph

A vertex-labeled graph is a graph $G = (V, A)$ with a label on the vertices $K \rightarrow \{0, ..., nc\}$, where nc is the number of different labels. This graph type has been employed in several domains such as biology (Zheng et al., 2011) or NLP (Bruckner et al., 2013). In this last study, the correction of Wikipedia inter-language links was modeled as a Colorful Components problem. Given a vertex-colored graph, the Colorful Components problem aims at finding the minimum-size edge sets that are connected and do not have two vertices with the same color.

In the context of MSC, we want to generate a short informative compression where keyword may be represented by several nodes in the word graph. Labels enable us to represent keywords in vertex-labeled graphs and generate a compression without repeated keywords while preserving the informativeness. In this framework we grant bonuses only once for nodes with the same label to prioritize new information in the compression. To make our model coherent, we added a base label (label 0) for all non-keywords in the word graph. The following section describes our ILP model to select sentences inside WGs by taking into account 3-grams and labeled keywords.

3.3 ILP Modeling

We denote K as the set of labels, each representing a keyword, and A as the set of arcs in the WG. T is defined as the set of the 3-grams occurring more than once.

There are several algorithms with a polynomial complexity to find the shortest path in a graph. However, the restriction on the minimum number P_{min} of vertices (i.e. the minimum number of words in the compression) makes the problem NP-hard. Indeed, let v_0 be the "begin" vertex. If P_{min} equals $|V|$ and if we add an auxiliary arc from "end" vertex to v_0, our problem is similar to the Traveling Salesman Problem (TSP), which is NP-Hard.

For this work we use the formulation known as Miller-Tucker-Zemlin (MTZ) to solve our problem (Öncan et al., 2009; Thadani and McKeown, 2013). This formulation uses a set of auxiliary variables, one for each vertex in order to prevent a vertex from being visited more than once in the cycle and a set of arc restrictions.

The problem of production of a compression that favors informativeness and grammaticality is expressed as Equation 6. In other words, we look for a path (sentence) that has a good cohesion and contains a maximum of labels (keywords) and relevant 3-grams.

$$\min \left(\sum_{(i,j) \in A} w(i,j) \cdot x_{i,j} - c \cdot \sum_{k \in K} b_k - \sum_{t \in T} d_t \cdot z_t \right) \quad (6)$$

where x_{ij} indicates the existence of the arc (i,j) in the solution, $w(i,j)$ is the arc weight between the words i and j (Equation 1), z_t indicates the existence of the 3-grams t in the solution, d_t is the relevance of the 3-grams t (Equation 4), b_k indicates the existence of a word with label (keyword) k in the solution and c is the keyword bonus of the graph[3].

3.4 Structural Constraints

We describe the structural constraints for the problem of consistency in compressions and define the bounds of the variables. First, we consider the problem of consistency which requires an inner and an outer arc active for every word used in the solution, where y_v indicates the existence of the vertex v in the solution.

$$\sum_{i \in \delta^+(v)} x_{vi} = y_v \qquad \forall v \in V, \quad (7)$$

$$\sum_{i \in \delta^-(v)} x_{iv} = y_v \qquad \forall v \in V. \quad (8)$$

The second requirement for consistency associates 3-grams and 2-grams variables. We have a 3-gram (a, b, c) only if the 2-grams (a, b) and (b, c) are used.

[3] The keyword bonus allows the generation of longer compressions that are more informative.

$$2z_t \leq x_{ij} + x_{jl}, \qquad \forall t = (i,j,l) \in T. \tag{9}$$

The constraint (10) controls the minimum number of vertices (P_{min}) used in the solution.

$$\sum_{v \in V} y_v \geq P_{\text{min}}. \tag{10}$$

The set of constraints (11) matches label variables (keywords) with vertices (words), where $V(k)$ is the set of all vertices with label k.

$$\sum_{v \in V(k)} y_v \geq b_k, \qquad \forall k \in K. \tag{11}$$

Equality (12) sets the vertex v_0 in the solution.

$$y_0 = 1. \tag{12}$$

The restrictions (13) and (14) are responsible for the elimination of sub-cycles, where u_v ($\forall v \in V$) are auxiliary variables for the elimination of sub-cycles and M is a large number (e.g. $M = |V|$).

$$u_0 = 1, \tag{13}$$

$$u_i - u_j + 1 \leq M - M \cdot x_{ij} \quad \forall (i,j) \in A, j \neq 0. \tag{14}$$

Finally, equations (15) – (18) define the field of variables.

$$\begin{aligned} x_{ij} &\in \{0,1\}, & \forall (i,j) &\in A, & (15)\\ z_t &\in \{0,1\}, & \forall t &\in T, & (16)\\ y_v &\in \{0,1\}, & \forall v &\in V, & (17)\\ u_v &\in \{1,2,\ldots,|V|\}, & \forall v &\in V. & (18) \end{aligned}$$

We calculate the 50 best solutions according to the objective (6) having at least 8 words and at least one verb. Specifically, we find the best solution, then we add a constraint in the model to avoid this solution and repeat this process 50 times to find the other solutions.

The optimized score (Equation 6) does not explicitly take into account the size of the generated sentence. Contrary to Filippova's method, sentences may have a negative score because we subtract from the cohesion value of the path the introduced scores for keywords and 3-grams. Therefore, we use the exponential function to ensure a score greater than zero. Finally, we select the sentence with the lowest final score (Equation 19) as the best compression.

$$\text{score}_{norm}(s) = \frac{e^{\text{score}_{opt}(s)}}{|s|}, \tag{19}$$

where $\text{score}_{opt}(s)$ is the score of the sentence s from Equation 6.

4 Experimental Setup

Algorithms were implemented using the Python programming language with the takahe[4] and gensim[5] libraries. The mathematical model was implemented in C++ with the Concert library and we used the solver CPLEX 12.6[6].

We define the keyword bonus as the geometric average[7] of all arc weights in the graph.

4.1 Evaluation Datasets

Various corpora have been developed for MSC and are composed of clusters of similar sentences from different source news in English, French, Spanish or Vietnamese languages. The corpora used by Filippova (2010) and Boudin and Morin (2013) contain clusters of at least 7 or 8 similar sentences, whereas the data of McKeown et al. (2010) and Luong et al. (2015) have clusters limited to pairs of sentences. McKeown et al. (2010) collected 300 English sentence pairs taken from newswire clusters using Amazon's Mechanical Turk. Like this previous study, Luong et al. (2015) used the same experimental setup to accumulate 115 Vietnamese clusters with 2 sentences by group. Boudin and Morin (2013), McKeown et al. (2010) and Luong

[4] http://www.florianboudin.org/publications.html

[5] https://radimrehurek.com/gensim/models/ldamodel.html

[6] CPLEX is available at: https://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-community-edition/

[7] Each WG has different weight arcs, so it is important that keyword bonus has a correct value to allow the generation of slightly longer compressions. We tested several metrics (fixed values, arithmetic average, median, and geometric average of weights arcs of WG) to define the keyword bonus of WG and empirically found that the geometric average outperformed the others.

et al. (2015) made their corpora publicly available but only the corpus of Boudin and Morin (2013) is more suited to multi-document summarization or question-answering because the documents to analyze are usually composed of many similar sentences. Therefore, we use this corpus made of 618 French sentences spread over 40 clusters. Each cluster has 3 sentences compressed by native speakers, references having a compression rate of 60%.

4.2 Automatic and Manual Evaluations

The most important features of MSC are informativeness and grammaticality. Informativeness measures how informational is the generated text. As references are assumed to contain the key information, we calculated informativeness scores counting the n-grams in common between the compression and the reference compressions using the ROUGE system (Lin, 2004). In particular, we used the F-measure metrics ROUGE-1 and ROUGE-2, F-measure being preferred to recall for a fair comparison of various lengths of compressed sentences. Like in (Boudin and Morin, 2013), ROUGE metrics are calculated with stopwords removal and French stemming[8].

Due to limitations of ROUGE systems that only analyze 1-grams and 2-grams, we also led a manual evaluation with 5 French native speakers. The native speakers evaluated the compression in two aspects: informativeness and grammaticality. In the same way as (Filippova, 2010; Boudin and Morin, 2013), the native speakers evaluated the grammaticality in a 3-point scale: 2 points for a correct sentence; 1 point if the sentence has minor mistakes; 0 point if it is none of the above. Like grammaticality, informativeness is evaluated in the same range: 2 points if the compression contains the main information; 1 point if the compression misses some relevant information; 0 point if the compression is not related to the main topic.

5 Experimental Assessment

Compression rates are strongly correlated with human judgments of meaning and grammaticality (Napoles et al., 2011). On the one hand, too short compressions may compromise sentence structure, reducing the informativeness and grammaticality. On the other hand, longer compressions are more interesting for ATS when informativeness and grammaticality are decisive features. Consequently, we generate two kinds of compressions according to the number of keywords in the graph (5 or 10), which acts on the final size of the output sentences. The result tables are split into two parts, each having similar CRs and calculated from LSI, LDA or TextRank methods to identify the keywords of the clusters.

5.1 Results

Table 1 describes the results for the French corpus using ROUGE. The first two lines display the evaluation of the two baseline systems; the ROUGE scores measured with our method using either 5 or 10 keywords are shown in the next three lines and the last three lines respectively.

Globally, our ILP method outperforms both baselines according to ROUGE F-measures, but mostly using 10 keywords with higher CRs. The use of LDA and LSI to determine keywords gives better results than TextRank. ILP+LDA.5 and ILP+LSI.5 were better than the baselines for ROUGE-1 but the second baseline system generated shorter sentences with a better ROUGE-2 score.

We led a further manual evaluation to study the informativeness and grammaticality of compressions. We measured inter-rater agreement on the judgments we collected, obtaining the value of Fleiss' kappa of 0.303. This result shows that human evaluation is rather subjective. Questioning evaluators on how they proceed to rate sentences reveals that they often made their choice by comparing outputs for a given cluster.

Table 1 also shows the manual analysis that ratifies the good results of our system. Informativeness scores are consistently improved by the ILP method, whereas grammaticality results measured on the three systems are similar. Filippova's method obtained the highest value for grammatical quality. However, our system led to informativeness scores better than the two baselines using 5 and 10 keywords.

Finally, the reranking method proposed by Boudin and Morin, based on the analysis of *keyphrases* of candidate compressions with TextRank, improves informativeness, but not to the same degree as our ILP model. Despite this gain, the method is limited to candidate sentences generated by Filippova's and is dependent on the TextRank method.

[8] http://snowball.tartarus.org/

Methods	Automatic Evaluation		Informativeness				Grammaticality			
	ROUGE-1	ROUGE-2	0	1	2	Avg.	0	1	2	Avg.
Filippova	0.6383[7]	0.4423[7]	11%	55%	34%	1.23	1%	26%	73%	***1.72***
Boudin et al.	0.6595[7]	*0.4616*[7]	6%	56%	38%	1.32	1%	31%	68%	1.68
ILP+LDA.5	0.6591	0.4383	11%	46%	43%	1.33	4%	24%	72%	1.67
ILP+LSI.5	*0.6615*	0.4368	9%	49%	42%	*1.34*	3%	28%	69%	1.65
ILP+TR.5	0.6576	0.4382	9%	54%	37%	1.28	4%	27%	69%	1.64
ILP+LDA.10	**0.6871**	0.4712	7%	45%	48%	**1.40**	2%	34%	64%	1.62
ILP+LSI.10	0.6862	**0.4713**	8%	43%	49%	**1.40**	2%	33%	65%	1.63
ILP+TR.10	0.6495	0.4355	10%	51%	39%	1.28	6%	33%	61%	1.54

Table 1: ROUGE results and manual evaluations on the French corpus. The results in italics represent the best results with CR closed to the baselines. The best ROUGE results are in bold.

5.2 Discussion

The measures done with both the automatic metrics ROUGE and human evaluations bring to light that the method used to select keyword acts on the performance of our ILP method. We investigated this through the analysis of selected keywords occurring in one of the reference compressions (LDA_5: 91%, LDA_10: 84%, LSI_5: 90%, LSI_10: 84%, TextRank_5: 69%, TextRank_10: 56%). As expected, a higher percentage of keywords can be found in the references when the top 5 instead of 10 are considered. In keeping with the performance evaluations, a significantly higher rate of keywords existing in the references is observed when using LDA or LSI rather than TextRank. This shows the prominent role of keyword selection for our MSC method. Most keywords identified by LDA and LSI methods are the same (around 91%) while the intersection of keywords between LDA and TextRank methods is around 50% (a similar level is measured for the intersection from LSI and TextRank). This overlap of keywords justifies the similar results produced by our method using LDA and LSI methods.

Short compressed sentences are appropriate to summarize documents; however, they may remove key information and prejudice the informativeness of the compression. For instance, for the sentences that would be associated with a higher relevant score by the ATS system, producing longer sentences would be more appropriate. Generating longer sentences makes easier to keep informativeness but often increases difficulties to have a good grammatical quality while combining different parts of sentences. The experimental results we presented in Section 5.1 indicate that scores on 3-grams provide a good stability for our method to generate grammatically correct sentences, even for longer compressions.

The length of the size of the sentences output by our ILP method can evolve as needed in two ways. Firstly, our method can ensure to keep enough information, through the number of considered keywords. Increasing this number generates longer compressions because our method tries to add more keywords. Table 2 presents the average size of compressions according to the used MSC setting. Globally, the size is increased by 2 words when using the second baseline with respect to the first one. Our ILP system generates sentences of the same size as the second baseline when labeling 5 keywords in WG and compressions longer by 2 when labeling 10 keywords, which is still a moderate increase. Moreover, Table 2 displays the number of keywords that are kept in the final compression and shows that for TextRank, less competitive than LDA and LSI, several keywords are discarded by the ILP score that also takes into account cohesion and 3-grams scores.

Secondly, our ILP model can include an explicit constraint over the compression size ($MaxSize$):

$$\sum_{v \in V} y_v \leq MaxSize. \tag{20}$$

[8] Although we used the same system and data as Boudin and Morin (2013) for the French corpus, we were not able to exactly reproduce their results. The ROUGE scores given in their article are close to ours for their system: 0.6568 (ROUGE-1) and 0.4414 (ROUGE-2), but using Filippova's system we measured higher scores than them: 0.5744 (ROUGE-1) and 0.3921 (ROUGE-2). In spite of these discrepancies, both ROUGE scores and manual evaluations (Table 1) led to the same conclusions as them, showing that their method outperform Filippova's.

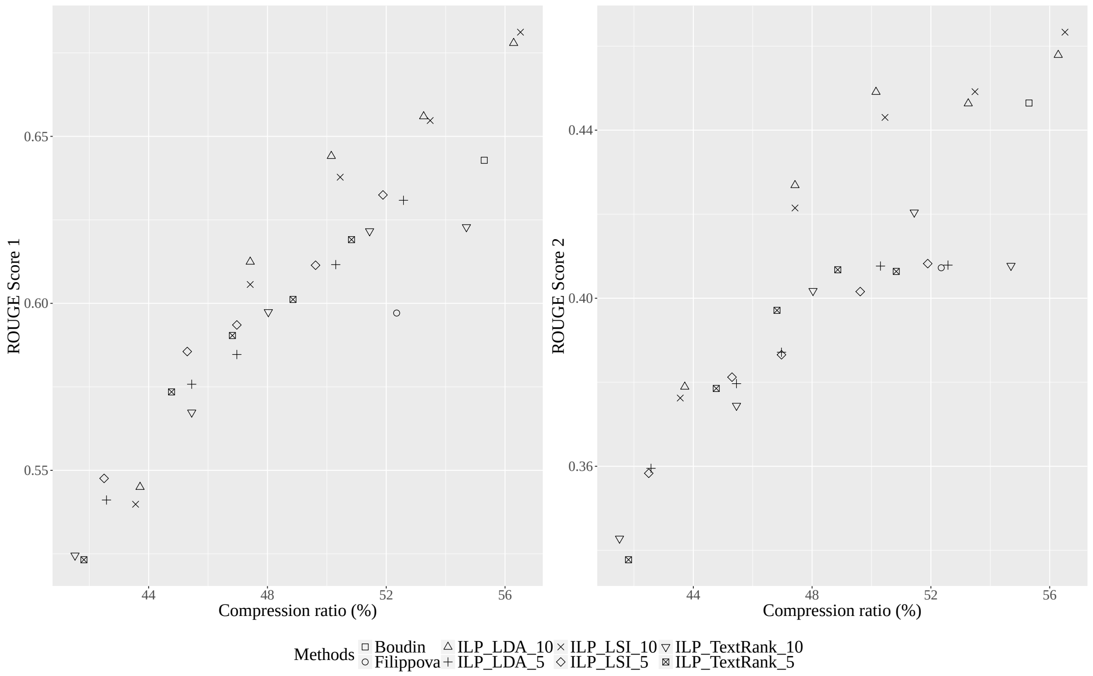

Figure 1: ROUGE recall for different maximum compression lengths using the French corpus.

Methods	**Length**		**Keywords**	**CR**
	Avg.	**Std.Dev.**		
Filippova	16.9	5.1	—	52%
Boudin et al.	18.3	4.9	—	55%
ILP+LDA.5	18.7	7.0	4.6	56%
ILP+LSI.5	18.8	7.1	4.6	57%
ILP+TR.5	18.1	6.9	3.2	55%
ILP+LDA.10	20.7	6.7	8.2	62%
ILP+LSI.10	21.1	6.5	8.4	64%
ILP+TR.10	20.0	7.6	5.5	60%

Table 2: Compression length (#words), standard deviation and number of used keywords computed on the French corpus.

We set up our system to generate compressions with average lengths of 55%, 60%, 65%, 70% and 75% and report the results measured in terms of ROUGE with each setting in Figure 1. Unlike Table 1, we measure ROUGE recalls instead of ROUGE F-measures because these first metrics have a better correlation with informativeness and we already take into account the size of the produced sentences through CR.

First, let us note that the CRs effectively observed may differ from the fixed value of $MaxSize$. For example, a 55% threshold leads to real CRs of 42% to 44%, while a 65% level creates new sentences with a real CR between 47% and 51%. Interestingly, our system obtained better ROUGE recall scores than both baselines for comparable compression lengths. If we prioritize meaning, our method with 10 keywords improved the compression quality with a small increase of the compression length (compression ratio between 60% and 64%). Instead, we can limit the length of compressions and generate compressions that are shorter and have still better ROUGE scores than the baselines.

6 Conclusion

Multi-Sentence Compression aims to generate a short informative text summary from several sentences with related and redundant information. Previous works built word graphs weighted by cohesion scores from the input sentences, then selected the best path to select words of the output sentence. We introduced in this study a model for MSC with two novel features. Firstly, we extended the work done by Boudin and Morin (2013) that introduced keywords to post-process lists of N-best compressions. We proposed to represent keywords as labels directly on the vertices of word graphs to ensure the use of different keywords in the selected paths. Secondly, we presented a new relevance score for 3-grams to maintain grammaticality. We devised an ILP modeling to take into account these two new features with the cohesion scores, while selecting the best sentence. The compression ratio can be modulated with this modeling, by selecting for example a higher number of keywords for the sentences considered essential for a summary. Automatic measures with ROUGE package were supplemented with a manual evaluation carried out by human judges in terms of informativeness and grammaticality. We showed that keywords and relevant 3-grams are important features to produce valuable compressed sentences; in particular, testing three different keyword selectors highlighted their role in producing relevant sentences. The paths selected with theses features generate results consistently improved in terms of informativeness while keeping up their grammaticality.

There are several potential avenues of work. Following the system proposed by ShafieiBavani et al. (2016), language models over POS can be added as an additional score to the optimization criterion to improve grammaticality. Another objective can be to manage polysemy through the use of the same label for the synonyms of each keyword inside the Word Graph. Finally, MSC can be jointly employed with classical methods of Automatic Text Summarization by extraction in order to generate better summaries.

References

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ILP based multi-sentence compression. In *IJCAI*. pages 1208–1214.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics* 31(3):297–328.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022.

Florian Boudin and Emmanuel Morin. 2013. Keyphrase extraction for N-best reranking in multi-sentence compression. In *NAACL*. pages 298–305.

Sharon Bruckner, Falk Hüffner, Christian Komusiewicz, and Rolf Niedermeier. 2013. *Evaluation of ILP-Based Approaches for Partitioning into Colorful Components*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 176–187.

James Clarke and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *EMNLP-CoNLL*.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *JAIR* 41:399–429.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for Information Science* 41(6):391–407.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *COLING*. pages 322–330.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *EMNLP*. ACL, pages 360–368.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *EMNLP*. pages 177–185.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Workshop Text Summarization Branches Out (ACL'04)*. pages 74–81.

A. V. Luong, N. T. Tran, V. G. Ung, and M. Q. Nghiem. 2015. Word graph-based multi-sentence compression: Re-ranking candidates using frequent words. In *7th International Conference on Knowledge and Systems Engineering (KSE)*. pages 55–60.

Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *HLT-NAACL*. pages 317–320.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *EMNLP*. pages 319–328.

R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into texts. In *EMNLP*.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Workshop on Monolingual Text-To-Text Generation (MTTG)*. pages 91–97.

Temel Öncan, İ Kuban Altınel, and Gilbert Laporte. 2009. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research* 36(3):637–654.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*. pages 379–389.

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond K. Wong, and Fang Chen. 2016. An efficient approach for multi-sentence compression. In Robert J. Durrant and Kee-Eung Kim, editors, *8th Asian Conference on Machine Learning*. PMLR, Hamilton, NZ, volume 63 of *Machine Learning Research*, pages 414–429.

Kapil Thadani and Kathleen McKeown. 2013. Supervised sentence fusion with single-stage inference. In *IJCNLP*. pages 1410–1418.

Juan-Manuel Torres-Moreno. 2014. *Automatic Text Summarization*. John Wiley & Sons.

Emmanouil Tzouridis, Jamal Nasir, and Ulf Brefeld. 2014. Learning to summarise related sentences. In *COLING, Technical Papers*. pages 1636–1647.

Chunfang Zheng, Krister Swenson, Eric Lyons, and David Sankoff. 2011. OMG! orthologs in multiple genomes — competing graph-theoretical formulations. In *WABI*. pages 364–375.