On Improving Informativity and Grammaticality for Multi-Sentence Compression

Elahe Shafiei Mohammad Ebrahimi Raymond Wong

Fang Chen

University of New South Wales, Australia
ATP Laboratory, National ICT, Sydney, Australia
{elahehs,mohammade,wong,fang}@cse.unsw.edu.au

Technical Report
UNSW-CSE-TR-201517
November 2015

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Multi Sentence Compression (MSC) is of great value to many real world applications, such as guided microblog summarization, opinion summarization and newswire summarization. Recently, word graph-based approaches have been proposed and become popular in MSC. Their key assumption is that redundancy among a set of related sentences provides a reliable way to generate informative and grammatical sentences. In this paper, we propose an effective approach to enhance the word graph-based MSC and tackle the issue that most of the state-of-the-art MSC approaches are confronted with: i.e., improving both informativity and grammaticality at the same time. Our approach consists of three main components: (1) a merging method based on Multiword Expressions (MWE); (2) a mapping strategy based on synonymy between words; (3) a re-ranking step to identify the best compression candidates generated using a POS-based language model (POS-LM). We demonstrate the effectiveness of this novel approach using a dataset made of clusters of English newswire sentences. The observed improvements on informativity and grammaticality of the generated compressions show that our approach is superior to state-of-the-art MSC methods.

1 Introduction

Multi-Sentence Compression (MSC) refers to the method of mapping a collection of related sentences to a sentence shorter than the average length of the input sentences, while retaining the most important information that conveys the gist of the content, and still remain grammatically correct [16, 4]. MSC is one of the challenging tasks in natural language processing that has recently attracted increasing interest [4]. This is mostly because of its potential use in various applications such as guided microblog summarization, opinion summarization, newswire summarization, text simplification for mobile devices and so on. A standard way to generate summaries usually consists of the following steps: ranking sentences by their importance, clustering them by similarity, and selecting a sentence from the top ranked clusters [31].

Traditionally, most of the MSC approaches rely on syntactic parsers, e.g. [10, 8]. As an alternative, some recent works in this field [9, 4] are based on word graphs, which only require a Part-Of-Speech (POS) tagger and a list of stopwords. These approaches simply rely on the words of the sentences and efficient dynamic programming. They take advantage of the redundancy among a set of related sentences to generate informative and grammatical sentences.

Although the proposed approach in [9] introduces an elegant word graph to MSC, approximately half of their generated sentences are missing important information about the set of related sentences [4]. Afterwards, Boudin and Morin (2013) enhanced their work and produced more informative sentences by maximizing the range of topics they cover. However, they confirmed that grammaticality scores are decreased, since their re-ranking algorithm produces longer compressions to ameliorate informativity. Therefore, grammaticality might be sacrificed while enhancing informativity and vice versa.

In this paper, we are motivated to tackle the main difficulty of the above mentioned MSC approaches which is to simultaneously improve both informativity and grammaticality of the compressed sentences. To this end, we propose a novel enhanced word graph-based MSC approach by employing significant merging, mapping and re-ranking steps that favor more informative and grammatical compressions. The contributions of the proposed method can be summarized as follows: (1) we exploit Multiword Expressions (MWE) from the given sentences and merge their words, constructing each MWE into a specific node in the word graph to reduce the ambiguity of mapping, so that well-organized and more informative compressions can be produced; (2) we take advantage of the concept of synonymy in two ways: firstly, we replace a merged MWE with its *one*-word synonym if available, and secondly, we use the synonyms of an upcoming single word to find the most proper nodes for mapping; (3) we employ a 7-gram POS-based language model (POS-LM) to re-rank the k-shortest obtained paths, and produce well-structured and more grammatical compressions. To our knowledge, this paper presents the first attempt to use MWEs, synonymy and POS-LM to improve the quality of word graph-based MSC. Extensive experiments on the released standard dataset demonstrate the effectiveness of our proposed approach. Figure 1.1 also depicts the overview of this approach.

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 presents our proposed approach. The data preparation process for evaluating our method is demonstrated in Section 4, and Section 5 reports the evaluation metrics and the performed experiments. Finally, Section

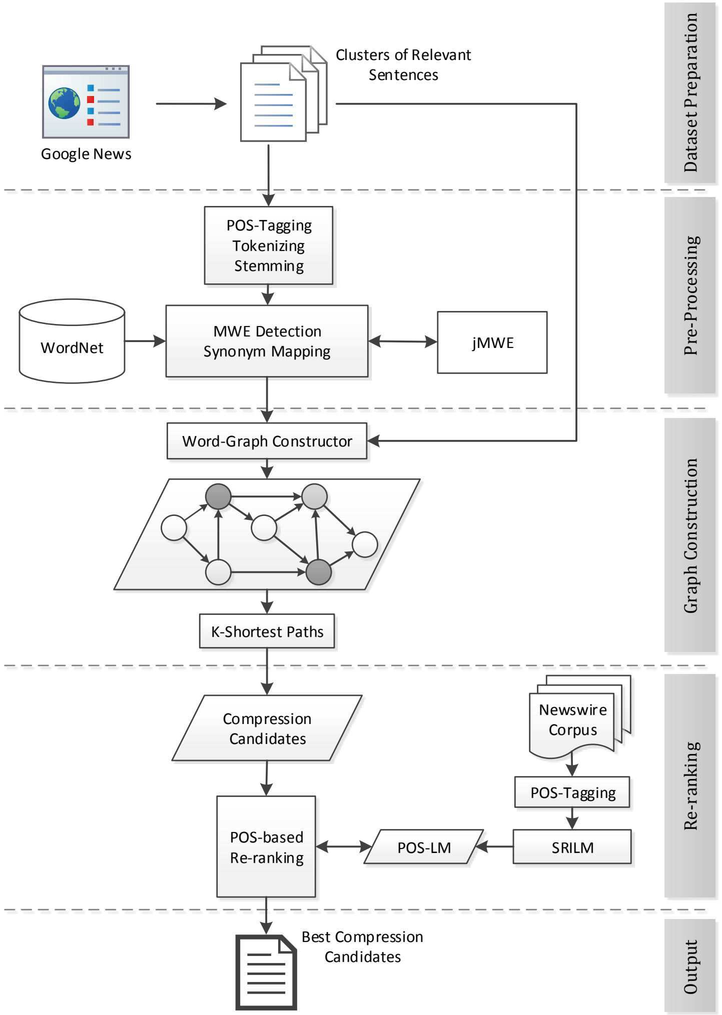


Figure 1.1: Overview of the proposed approach

6 concludes the paper.

2 Related Work

2.1 Multi-Sentence Compression

State-of-the-art approaches in the field of MSC are generally divided into supervised [21, 11] and unsupervised groups [6]. MSC methods traditionally use a syntactic parser to generate grammatical compressions, and fall into two categories (based on their implementations): (1) *tree-based* approaches, which create a compressed sentence by making edits to the syntactic tree of the original sentence [21, 11, 10, 8]; (2) *sentence-based* approaches, which generates strings directly [6].

As an alternative, word graph-based approaches that only require a POS tagger have recently been used in different tasks, such as guided microblog summarization [27], opinion summarization [12] and newswire summarization [9, 4, 30]. In these approaches, a directed word graph is constructed in which nodes represent words while edges between two nodes represent adjacency relations between words in a sentence. Hence, the task of sentence compression is performed by finding the k-shortest paths in the word graph. In particular, our work is applied to newswire summarization. In this field, Filippova (2010) has introduced an elegant word graph-based MSC approach that relies on the redundancy among the set of related sentences. However, some important information are missed from 48% to 60% of the generated sentences in their approach [4]. Thus, Boudin and Morin (2013) proposed an additional re-ranking scheme to identify summarizations that contain key phrases. However, they mentioned that grammaticality is sacrificed to improve informativity in their work.

In our proposed approach, we utilize MWEs and synonym words in sentences to significantly enhance the traditional word graph, and improve informativity. Then, we re-rank the generated compression candidates with a 7-gram POS-LM that captures the syntactic information, and strengthens the compressed sentences in terms of grammaticality.

2.2 Multiword Expressions

An MWE is a combination of words with lexical, syntactic or semantic idiosyncrasy [26, 3]. It is estimated that the number of MWEs in the lexicon of a native speaker of a language has the same order of magnitude as the number of single words [15]. Hence, explicit identification of MWEs has been shown to be useful in various NLP applications. Components of an MWE can be treated as a single unit to improve the effectiveness of re-ranking steps in IR systems [1]. In this paper, we identify MWEs, merge their components, and replace them with their available *one*-word synonyms, if applicable. These strategies help to construct an improved word graph and enhance the informativity of the compression candidates.

2.3 POS-based Language Model (POS-LM)

A language model assigns a probability to a sequence of m words $P(w_1, ..., w_m)$ by means of a probability distribution. Language models are an essential element of natural language processing, in tasks ranging from spell-checking to machine translation. Given the increasing need to ensure grammatical sentences in different applications, POS-LM comes into play as a remedy. POS-LM describes the probability of a sequence of m POS tags $P(t_1, ..., t_m)$. POS-LMs are traditionally used for speech recognition problems [14] and statistical machine translation systems [17, 23, 25] to capture syntactic information. In this paper, we benefit from POS-LMs to capture the syntactic information of sentences and improve the grammaticality of compression candidates.

3 Proposed Approach

3.1 Word Graph Construction for MSC

Consider a set of related sentences $S = \{s_1, s_2, ..., s_n\}$, a traditional word graph is constructed by iteratively adding sentences to it. This directed graph is an ordered pair $G = (V, E)$ comprising of a set of vertices or words together with a set of directed edges which shows the adjacency between corresponding nodes [9, 4]. The graph is firstly constructed by the first sentence and displays words in a sentence as a sequence of connected nodes. The first node is the start node and the last one is the end node. Words are added to the graph in three steps of the following order: (1) non-stopwords for which no candidate exists in the graph; or for which an unambiguous mapping is possible (i.e. there is only one node in the graph that refer to the same word/POS pair); (2) non-stopwords for which there are either several possible candidates in the graph; or for which they occur more than once in the sentence; (3) stopwords. For the last group, same as Boudin and Morin (2013), we use the stopword list included in nltk[1] extended with temporal nouns such as 'yesterday', 'Friday', and etc..

All MSC approaches aim at producing condensed sentences that inherit the most important information from the original content while remains syntactically correct. However, gaining these goals at the same time remains still difficult. As a remedy, we believe that a better resolution to construct an improved word graph can be obtained by using more sophisticated pre-processing and re-ranking steps. Thus, we focus on the notions of synonymy, MWE and POS-LM re-ranking, which dramatically raise the informativity and grammaticality of compression candidates. In the following, we describe the details of our proposed approach:

3.2 Merging and Mapping Strategies

Like many NLP applications, MSC will benefit from the identification of MWEs and the concept of synonymy; and even more so when lexical diversity arises in a collection of sentences. For example, consider a sentence that includes an MWE (*kick the bucket*): *It would be a sad thing to kick the bucket without having been to Alaska.* To benefit from this MWE that has 3 components/words, we propose the merging strategy below:

Firstly, after tokenizing the sentence and stemming the words, we detect the MWE and its tuple POS with an MWE detector. This step has the advantage of reducing the ambiguity of mapping upcoming words onto the existing words with the same appearance in the graph. For example, the word *kick* above has a different meaning and POS (as an MWE component) from the identical appearance word *kick* in isolation (in another sentence say they *kick* open the door and entered the room.). So, MWE identification can keep us from mapping these two *kick* together and retain the important meaning of the content. To detect MWEs, we use the jMWE toolkit [19], which is a Java-based library for constructing and testing MWE token detectors.

Secondly, we use version 3.0 of WordNet [22] to obtain its available *one*-word synonym with an appropriate POS and replace the n-words MWE with a shorter synonym word. WordNet groups all synonyms into a SynSet - a synonym set.

[1] http://nltk.org/

We only consider the most frequent *one*-word synonym in the WordNet that also appears in the other relevant sentences. If other relevant sentences contain none of the *one*-word synonyms, the most frequent one is selected directly from the WordNet to help condense the sentence. Three native speakers were asked to investigate all the synonym mappings performed in our approach, and specify whether each mapped synonym reflects the meaning of the original word in the sentence or not. Based on this evaluation, the average rate of correct synonym mappings is 88.21%. In case that no appropriate synonym is found for MWE, the merged MWE itself was used as a back-off. This can reduce the number of graph nodes and, consequently, the ambiguity for further false mappings of MWE components in the word graph. These steps are briefly depicted in Figure 3.1 (a).

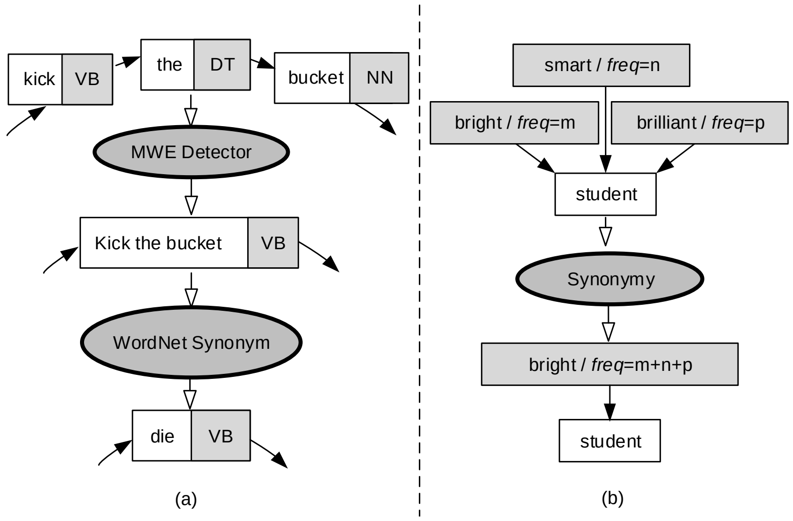


Figure 3.1: (a) Example of MWE merging and mapping, (b) Example of Synonym mapping

Furthermore, we use the concept of synonymy for mapping upcoming single words. For example, consider 3 different sentences containing words *bright*, *smart* and *brilliant*, which are synonyms of each other. Assume each sentence contains one of these synonyms respectively. Without an appropriate mapping based on a notion of synonymy, these 3 nodes will be added to the word graph as separate nodes. With our approach, the word graph in this example is constructed with a single node containing a word as a representative of its synonyms from the other sentences. The weight of the obtained node is computed by summing the frequency scores from the other nodes as shown in Figure 3.1 (b) for each pair of word/POS. The main purpose of this modification is three fold: (i) the ambiguity of mapping nodes is reduced; (ii) the number of total possible paths (compression candidates) is decreased; and (iii) the weight of frequent similar words with different appearances in the content is better reflected by the notion of synonymy.

In the following example, we will demonstrate how we use the pre-processing strategies to produce refined sentences, and generate an improved word graph.

Among the underlined words, MWEs are put into bracket, and synonyms are identified by the same superscript notations.

(1) Teenage^{a} boys are more interested^{b} in [junk food]^{c} marketing and consume^{d} more [fast food]^{c} than girls.
(2) [Junk food]^{c} marketers find young^{a} boys more fascinated^{b} than girls, a survey released^{e} by the Cancer Council shows.
(3) Adolescent^{a} boys [use up]^{d} more [fast food]^{c} than girls, [according to] a new survey.
(4) The survey, published^{e} by the Cancer Council, observed teenage^{a} boys were regular consumers of [junk food]^{c}.

The word graph constructed for the above sentences are partially shown in Figure 3.2. Some nodes, edge weights and punctuations are omitted from the graph for more clarity.

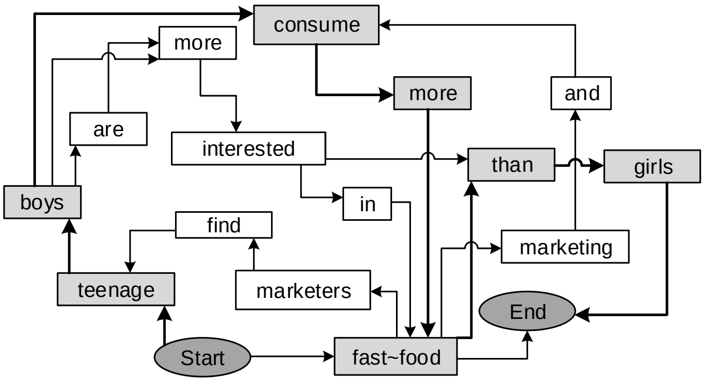


Figure 3.2: The generated word graph and a compression path

Where mapping in the graph is ambiguous (i.e. there are two or more nodes in the graph that refer to the same word/POS pair), we follow the instruction stated by Filippova (2010): the immediate context (the preceding and following words in the sentence, and the neighboring nodes in the graph) or the frequency (i.e. the node which has words mapped to it) is used to select the best candidate node. A new node is created only if there is no suitable candidate to be mapped to, in the graph.

In Filippova (2010), edge weights are calculated using the weighting function defined in Equation 3.1 in which $w^{'}(e_{i,j})$ is given by Equation 3.2.

$$w(e_{i,j}) = \frac{w^{'}(e_{i,j})}{freq(i) \times freq(j)} \tag{3.1}$$

$$w^{'}(e_{i,j}) = \frac{freq(i) + freq(j)}{\sum_{s \in S} diff(s,i,j)^{-1}} \tag{3.2}$$

where $freq(i)$ is the number of words mapped to the node i. The function $diff(s,i,j)$ refers to the distance between the offset positions of words i and j

in sentence s.

Algorithm 1 Proposed MSC Word Graph

```
1: Input: A cluster of relevant sentences: $S = \{s_i\}_{i=1}^{n}$
2: Output: $G = (V, E)$
3: for $i = 1$ to $n$ do
4:     $t \leftarrow Tokenize(s_i)$
5:     $st \leftarrow Stemming(t)$
6:     $MWE\text{-}comp \leftarrow MWE\text{-}Detection(t, st)$
7:     $MWE\text{-}list \leftarrow Merge\text{-}MWE(MWE\text{-}comp)$
8:     $sentSize \leftarrow SizeOf(t)$
9:     for $j = 1$ to $sentSize$ do
10:        $LABEL \leftarrow t_j$
11:        $SID \leftarrow i$
12:        $PID \leftarrow j$
13:        $SameN \leftarrow getSameNodes(G, LABEL)$
14:        if $sizeOf(SameN) \geq 1$ then
15:            $v_j \leftarrow getBestSame(SameN)$
16:            $mapList_{v_j} \leftarrow mapList_{v_j} \cup (SID, PID)$
17:        else
18:            $SynN \leftarrow getSynonymNodes(G, LABEL)$
19:            if $sizeOf(SynN) \geq 1$ then
20:                $v_j \leftarrow getBestSyn(SynN)$
21:                $mapList_{v_j} \leftarrow mapList_{v_j} \cup (SID, PID)$
22:            esle if $t_j \in MWE\text{-}list$ then
23:                $WNSyn \leftarrow getBestWNSyn(LABEL)$
24:                $v_j \leftarrow creatNewNode(G, WNSyn)$
25:                $mapList_{v_j} \leftarrow (SID, PID)$
26:            esle
27:                $v_j \leftarrow creatNewNode(G, LABEL)$
28:                $mapList_{v_j} \leftarrow (SID, PID)$
29:            end if
30:        end if
31:        if not $existEdge(G, v_{j-1} \rightarrow v_j)$ then
32:            $addEdge(v_{j-1} \rightarrow v_j, G)$
33:        end if
34:    end for
35: end for
```

Algorithm 1 presents the steps to build our proposed MSC word graph, G(V, E). We start with a cluster of relevant sentences from a set of input newswire clusters. Each cluster is denoted as $S = \{s_i\}_{i=1}^{n}$ where each s_i is a sentence containing POS annotations. *Line 4-5:* Each $s_i \in S$ is split into a set of tokens, where each token, t_j consists of a word and its corresponding POS annotation (e.g.*"boys:NN"*). The tokens are also stemmed into a set of stemmed words, *st*. *Line 6-7:* For each sentence, MWE components, i.e., *MWE-comp*, are detected using the set of tokens t and stems *st*. Then, these MWE components are merged in each sentence, and kept in a list of *MWE-list*. *Line 10-12:* Each unique t_j will form a node v_j in the MSC graph, with t_j being the label. Since we only have one node per unique token, each node keeps track of all sentences that include its token. So, each node keeps a list of *sentence identifier*, (SID) along with the *position of token* in that sentence, (PID). Each node including a single word or a merged MWE will thus carry a *mapping list* (mapList) which is a list of {SID:PID} pairs representing the node's membership in a sentence.

Line 13-16: For mapping the token t_j, we first explore the graph to find the same node (i.e. node that refers to the same word/POS pair as t_j). If two or more same nodes are found, considering the aforementioned ambiguous mapping criteria in Section 3.2, the best candidate node is selected for mapping. Then the pair of (SID:PID) of t_j will be added to the mapping list of the selected node, i.e., $mapList_{v_j}$. *Line 18-21:* If no same node exists in the graph, then we look for the best synonym node in the graph (i.e. find the most frequent synonym among the WordNet synsets that was earlier added to the graph.). Again, the mapping list of the selected node, $mapList_{v_j}$ will be updated to include the pair of (SID:PID) of t_j. *Line 22-28:* If none of the above conditions are satisfied, it is time to create a new node in the graph. However as explained in Section 3.2, when t_j is MWE, we extract the best WordNet *one*-word synonym, and replace the n-word MWE with this shorter synonym word. So, a shorter content node will be added to the graph. *Line 31-33:* the original structure of a sentence is reordered with the use of directed edges.

A heuristic algorithm is then used to find the k-shortest paths from start to end node in the graph. Throughout our experiments, the appropriate value for k is 150. By re-ranking this number of shortest paths, most of the potentially good candidates are kept and a decline in performance is prevented. Paths shorter than eight words or do not contain a verb are filtered before re-ranking. The remaining paths are re-ranked and the path that has the lightest average edge weight is eventually considered as the best compression. Next, an accurate re-ranking approach to identify the most informative grammatical compression candidate is described.

3.3 Re-ranking Strategy (POS-LM)

Boudin and Morin (2013) have recently utilized TextRank (Mihalcea and Tarau 2004) to re-rank the compression candidates. In their approach, a word recommends other co-occurring words, and the strength of the recommendation is recursively computed based on the importance of the words making the recommendation. The score of a keyphrase k is computed by summing the salience of the words it contains, normalized with its $length + 1$ to favor longer n-grams according to Equation 3.3.

$$score(k) = \frac{\sum_{w \in k} TextRank(w)}{length(k) + 1} \tag{3.3}$$

Finally, the paths are re-ranked and the score of a compression candidate c is given by Equation 3.4.

$$score(c) = \frac{\sum_{i,j \in path(c)} w(e_{i,j})}{length(c) \times \sum_{k \in c} score(k)} \tag{3.4}$$

In our re-ranking step, we benefit from the fact that POS tags capture the syntactic roles of words in a sentence. We use a POS-LM to assign a grammaticality score to each generated compression. Our hypothesis is that POS-LM helps in identifying the most grammatical sentence among the k-most informative compressions. This strategy shall improve the grammaticality of MSC, even when the grammatical structures of the input sentences are completely different. Word-based language models estimate the probability of a string of m words by Equation 3.5, and POS-LMs estimate the probability of string of m POS tags by Equation 3.6 [23].

$$p(w_1^m) \propto \prod_{i-1}^{m} p(w_i | w_{i-n+1}^{i-1}) \tag{3.5}$$

$$p(t_1^m) \propto \prod_{i-1}^{m} p(t_i | t_{i-n+1}^{i-1}) \tag{3.6}$$

where, n is the order of the language model, and w/t refers to the sub-sequence of words/tags from position i to j.

To build a POS-LM, we use the SRILM toolkit with modified Kneser-Ney smoothing [28], and train the language model on our POS annotated corpus. SRILM collects n-gram statistics from all n-grams occurring in a corpus to build a single global language model. To train our POS-LM, we need a POS-annotated corpus. In this regard, we make use of the Stanford POS tagger [29] to annotate the AFE sections of LDCs Gigaword corpus (LDC2003T05) as a large newswire corpus (∼170 M-words). Then, we remove all words from the pairs of words/POS in the POS annotated corpus.

Although the vocabulary of a POS-LM, which is usually ranging between 40 and 100 tags, is much smaller than the vocabulary of a word-based language model, there is still a chance in some cases of unseen events. Since modified Kneser-Ney discounting appears to be the most efficient method in a systematic description and comparison of the usual smoothing methods [13], we use this type of smoothing to help our language model.

The compression candidates also need to be annotated with POS tags. So, the score of each compression is estimated by the language model, based on its sequence of POS tags. Since factors like POS tags, are less sparse than surface forms, it is possible to create a higher order language models for these factors. This may encourage more syntactically correct output [18]. Thus, in our approach we use 7-gram language modeling based on part of speech tagging to re-rank the k-best compressions generated by the word graph.

To re-rank the obtained paths, our POS-LM gives the perplexity score ($score_{LM}$) which is the geometric average of *1/probability* of each sentence, normalized by the number of words. So, $score_{LM}$ for each sequence of POS in the k-best compressions is computed by Equation 3.7.

$$score_{LM}(c) = 10^{\frac{\log prob(c)}{\# word}} \tag{3.7}$$

where $prob(c)$ is the probability of compression *(C)* including $\#Word$ number of words, computed by the 7-gram POS-LM.

As the estimated scores for each cluster of sentences fall into different ranges, we make use of unity-based normalization to bring the values of $score(c)$ in Equation 4, and the $score_{LM}$ into the range [0, 1]. The score of each compression is finally given by Equation 3.8

$$score_{final}(c) = \mu \times score(c) + (1 - \mu) \times score_{LM}(c) \tag{3.8}$$

in which the scaling factor μ in our experiments has been set to 0.4, so as to reach the best re-ranking results.

To better understand how POS-LM is used, consider the sentences below, which have the same scores for informativity but are added into our re-ranking contest to be investigated based on their grammaticality. The corresponding POS sequences of these sentences are given to the trained language model to clarify which one is more grammatical.

(1) Boys *more* *consume* fast food than girls.
NNS $\underbrace{RBR \quad VBP}_{\text{Wrong Pattern}}$ JJ NN IN NNS

(2) Boys *consume* *more* fast food than girls.
NNS VBP JJR JJ NN IN NNS

As expected, the winner of this contest is the second POS sequence, which has a better grammatical structure and gets a higher probability score from the POS-LM.

4 Data Preparation

Many attempts have been made to release various kinds of datasets and evaluation corpora for sentence compression and automatic summarization, such as the one introduced in [5]. However, to our knowledge, there is no dataset available to evaluate MSC in an automatic way [4]. Since the prepared dataset in Boudin and Morin (2013) is also in French, we have followed the below instructions to construct a Standard English newswire dataset:

We have collected news articles in clusters on the Australian[1] and U.S.[2] edition of Google News over a period of five months (January 2015 - May 2015). Clusters composed of at least 15 news articles about one single news event, were manually extracted from different categories (i.e. Top Stories, World, Business, Technology, Entertainment, Science, Health, etc.). Leading sentences in news articles are known to provide a good summary of the article content and are used as a baseline in summarization [7]. Hence, to obtain the sets of related sentences, we have extracted the first sentences from the articles in the cluster and removed duplicates.

[1] http://news.google.com.au/

[2] http://news.google.com/

The released dataset contains 568 sentences spread over 46 clusters (each is related to one single news event). The average number of sentences within each cluster is 12, with a minimum of 7 and a maximum of 24. Three native English speakers were also asked to meticulously read the sentences provided in the clusters, extract the most salient facts, summarize the set of sentences, and generate three reference summaries for each cluster with as less new vocabularies as possible.

In practice, along with the clusters of sentences with similar lexical and grammatical structures (we refer to these clusters as *normal*), it is likely to have clusters of content-relevant sentences, but with different (non-redundant) appearances and grammatical structures (we consider these clusters as *diverse*). In fact, the denser a word graph is, the more edges interconnect with vertices and hence more paths pass through the same vertices. This results in low lexical and syntactical diversity, and vice versa [30]. The density of a word graph generated by sentences of a cluster $G = (V, E)$ is given by Equation 4.1.

$$Density = \frac{|E|}{|V|(|V| - 1)} \tag{4.1}$$

Thereupon, we have also identified 15 *diverse* clusters among the 46 clusters to demonstrate the effect of our approach on the normal and diverse groups. Table 4.1 lists the properties of the evaluation dataset.

total #clusters	46
#normal clusters	31
#diverse clusters	15
total #sentences	568
avg #sentences/cluster	12
min #sentences/cluster	7
max #sentences/cluster	24

Table 4.1: Information about the constructed dataset

5 Experiments

5.1 Evaluation Metrics

We evaluate the proposed method over our constructed dataset (*normal* and *diverse* clusters) using automatic and the manual evaluations. The quality of the generated compressions was assessed automatically through version 2.0 [1] of ROUGE [20] and the version 13a [2] of BLEU [24]. These sets of metrics are typically used for evaluating automatic summarization and machine translation. They compare an automatically produced summary against a reference or a set of human-produced summaries.

[1] http://kavita-ganesan.com/content/rouge-2.0

[2] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl

For the manual investigation of the quality of the generated compressions, three native English speakers were asked to rate the grammaticality and informativity of the compressions based on the points scale defined in Filippova (2010). *Grammaticality*: (i) if the compression is grammatically perfect → *point 2*; (ii) if the compression requires some minor editing → *point 1*; (iii) if the compression is ungrammatical → *point 0*. The lack of capitalization is ignored by the raters. *Informativity*: (i) if the compression conveys the gist of the content and is mostly similar to the human-produced summary → *point 2*; (ii) if the compression misses some important information → *point 1*; (iii) if the compression contains none of the important contents → *point 0* (Table 5.1).

The k value for the agreement between raters falls into range ($0.4 \sim 0.6$) through Kappa's evaluation metrics, which indicates that the strength of this agreement is moderate [2].

Feature	**State of the Compression**	**Point**		
		2	1	0
Grammaticality	grammatically perfect	√		
	requires some minor editing		√	
	ungrammatical			√
Informativity	conveys the gist of the content	√		
	misses some important information		√	
	contains none of the important contents			√

Table 5.1: Points scale defined in the agreement between raters

5.2 Experiment Results

Two existing approaches, i.e., Filippova (2010) and Boudin and Morin (2013) are used as Baseline1 and Baseline2 respectively, for comparison purposes in our experiments. To better understand the behavior of our system, we examined our test dataset, and made the following observations. For the manual evaluation (Table 5.2), we observed a significant improvement in the average grammaticality and informativity scores along with the compression ratio (CompR) over the normal and diverse clusters. The informativity of Baseline1 is adversely influenced by missing important information about the set of related sentences [4]. However Baseline2 enhanced the informativity, the grammaticality scores are decreased due to the outputs of longer compressions. In our approach, the remarkable improvement in the grammaticality scores is due to the adding of the syntactic-based re-ranking step. Using this re-ranking method, the most grammatical sentences are picked among the k-best compression candidates. Furthermore, merging MWEs, replacing them with their available *one*-word synonyms and mapping words using synonymy all enhance the informativity scores, and help to generate a denser word graph instead of a sparse one. Given that, the value of the compression ratio (∼48%) is better than the best obtained compression ratio on these two baselines (50%).

The average performance of the baseline methods and the proposed approach

Method	**Normal**		**Diverse**		**CompR**
	Info.	**Gram.**	**Info.**	**Gram.**	
Baseline1	1.44	1.67	1.17	1.19	50%
Baseline2	1.68	1.60	1.30	1.12	58%
Proposed	1.68	1.68	1.36	1.47	48%

Table 5.2: Average scores over normal and diverse clusters separately given by the raters; along with the estimated compression rate

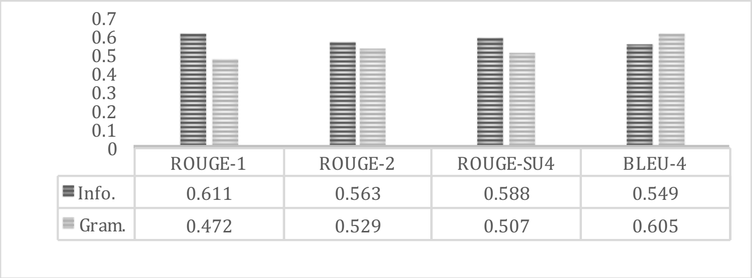
over the normal and diverse clusters in terms of Rouge and Bleu scores are also shown in Table 5.3. Rouge measures the concordance of candidate and reference summaries by determining n-gram, word sequence, and word pair matches. We used Rouge F-measure for unigram, bigrams, and su4 (skip-bigram with maximum gap length 4) to evaluate the compression candidates. The Bleu metric computes the scores for individual sentences; then averages these scores over the whole corpus for a final score. We used Bleu for 4-grams to evaluate the results.

Metric	**Baseline1**	**Baseline2**	**Proposed**
Rouge-1	0.4912	0.5093	0.5841
Rouge-2	0.3050	0.3131	0.4284
Rouge-su4	0.2867	0.3002	0.3950
Bleu-4	0.4510	0.5144	0.6913

Table 5.3: Average scores by automatic evaluation over the normal and diverse clusters

To make the candidate and reference summaries comparable, a process of manual MWE detection is performed on the reference summaries and the MWE components are merged by three native annotators. In details, automatic evaluation packages use WordNet to compare the synonyms in candidate and reference summaries. WordNet puts hyphenation on synonyms, e.g., kick-the-bucket, so annotators hyphenate MWEs in their summaries to be used in these packages. Then, the synonym properties are set in these packages to consider the synsets. Thus, n-words MWEs are linked to their *one*-word synonyms in the candidate summary. The overall results support our hypothesis that using the POS-LM for re-ranking the compression candidates, results in more grammatical compressions, especially for diverse clusters. This issue is confirmed by 4-grams Bleu, which shows the grammaticality enhancement rather than the informativity. Meanwhile, we try to simultaneously improve the informativity by identifying and merging MWEs along with mapping the synonyms.

Furthermore, the effectiveness of Rouge and Bleu is studied using the Pearson's correlation coefficient. We found that Rouge shows a better correlation with informativity, while the Bleu correlates better with grammaticality. Overall, the results in Figure 5.1 show high correlation ($0.5 \sim 1.0$) between the automatic evaluation results and human ratings for both Rouge and Bleu. The main reason may be the simulation of factors that humans usually consider for summarization, such as merging and mapping strategies, along with the syntactic criteria employed by POS-LM.



	ROUGE-1	ROUGE-2	ROUGE-SU4	BLEU-4
▨ Info.	0.611	0.563	0.588	0.549
▨ Gram.	0.472	0.529	0.507	0.605

Figure 5.1: The effectiveness of Rouge and Bleu

To investigate the impact of each improvement separately, we have also conducted separate experiments over the prepared dataset. The results are shown in Figure 5.2 and the related data are provided in Table 5.4. In our work, merging and mapping strategies significantly increase the informativity of the compressions. So, their computed scores by Rouge are higher than the score of POS-LM. However, the combination of MWE merging and mapping gets a slightly lower score from Rouge-su4. One reason may be that usage of synonymy only for MWEs and ignoring other *one*-word synonym mapping causes a more diverse graph, which slightly decreases the informativity and grammaticality of compressed sentences. Meanwhile, POS-LM gets better scores from Bleu-4, which indicates the grammaticality enhancement rather than the informativity.

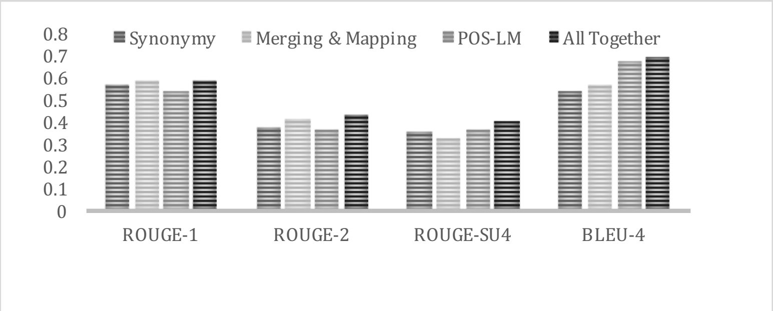


Figure 5.2: The effects of the improvements separately

6 Conclusions

In a nutshell, we have presented our attempt in using MWEs, Synonymy and POS-based language modeling to tackle one of the pain points of MSC, which is improving both informativity and grammaticality at the same time. By manual and automatic (Rouge and Bleu) evaluations, experiments using a constructed

Metric	Synonymy	Merg/Map	POS-LM	All
Rouge-1	0.5659	0.5820	0.5381	0.5841
Rouge-2	0.3723	0.4087	0.3599	0.4284
Rouge-su4	0.3508	0.3254	0.3629	0.3950
Bleu-4	0.5340	0.5601	0.6725	0.6913

Table 5.4: The effects of the improvements separately

English newswire dataset show that our approach outperforms the competitive baselines. In particular, the proposed merging and mapping strategies, along with the grammar-enhanced POS-LM re-ranking method, ameliorate both informativity and grammaticality of the compressions, with an improved compression ratio.

Bibliography

[1] Otavio Costa Acosta, Aline Villavicencio, and Viviane P Moreira. Identification and treatment of multiword expressions applied to information retrieval. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 101–109. Association for Computational Linguistics, 2011.

[2] Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008.

[3] Timothy Baldwin and Su Nam Kim. Multiword expressions. *Handbook of Natural Language Processing, second edition. Morgan and Claypool*, 2010.

[4] Florian Boudin and Emmanuel Morin. Keyphrase extraction for n-best reranking in multi-sentence compression. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013.

[5] James Clarke and Mirella Lapata. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 377–384. Association for Computational Linguistics, 2006.

[6] James Clarke and Mirella Lapata. Modelling compression with discourse constraints. In *EMNLP-CoNLL*, pages 1–11, 2007.

[7] Hoa Trang Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, pages 1–12, 2005.

[8] Micha Elsner and Deepak Santhanam. Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 54–63. Association for Computational Linguistics, 2011.

[9] Katja Filippova. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. Association for Computational Linguistics, 2010.

[10] Katja Filippova and Michael Strube. Sentence fusion via dependency graph compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 177–185. Association for Computational Linguistics, 2008.

[11] Michel Galley and Kathleen McKeown. Lexicalized markov grammars for sentence compression. In *HLT-NAACL*, pages 180–187, 2007.

[12] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

[13] Joshua T Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001.

[14] Peter A Heeman. Pos tagging versus classes in language modeling. In *Proceedings of the 6th Workshop on Very Large Corpora, Montreal*, 1998.

[15] Ray Jackendoff. *The architecture of the language faculty*. Number 28. MIT Press, 1997.

[16] Hongyan Jing. Sentence reduction for automatic text summarization. In *Proceedings of the sixth conference on Applied natural language processing*, pages 310–315. Association for Computational Linguistics, 2000.

[17] Philipp Koehn, Abhishek Arun, and Hieu Hoang. Towards better machine translation quality for the german–english language pairs. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 139–142. Association for Computational Linguistics, 2008.

[18] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[19] Nidhi Kulkarni and Mark Alan Finlayson. jmwe: A java toolkit for detecting multi-word expressions. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 122–124. Association for Computational Linguistics, 2011.

[20] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8, 2004.

[21] Ryan T McDonald. Discriminative sentence compression with soft syntactic evidence. In *EACL*, 2006.

[22] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[23] Christof Monz. Statistical machine translation with local language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 869–879. Association for Computational Linguistics, 2011.

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[25] Maja Popović. Morpheme-and pos-based ibm1 scores and language model scores for translation quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 133–137. Association for Computational Linguistics, 2012.

[26] Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15. Springer, 2002.

[27] Beaux Sharifi, Mark-Anthony Hutton, and Jugal K Kalita. Experiments in microblog summarization. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 49–56. IEEE, 2010.

[28] Andreas Stolcke et al. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*, 2002.

[29] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[30] Emmanouil Tzouridis, Jamal Abdul Nasir, LUMS Lahore, and Ulf Brefeld. Learning to summarise related sentences. In *The 25th International Conference on Computational Linguistics (COLING14), Dublin, Ireland, ACL*, 2014.

[31] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM, 2008.