Cross-Language Text Summarization Using Sentence and Multi-Sentence Compression

Elvys Linhares Pontes[1](✉), Stéphane Huet[1], Juan-Manuel Torres-Moreno[1,2], and Andréa Carneiro Linhares[3]

[1] LIA, Université d'Avignon et des Pays de Vaucluse, 84000 Avignon, France
elvys.linhares-pontes@alumni.univ-avignon.fr
[2] École Polytechnique de Montréal, Montreal, Québec, Canada
[3] Universidade Federal do Ceará, Sobral, Ceará, Brazil

Abstract. Cross-Language Automatic Text Summarization produces a summary in a language different from the language of the source documents. In this paper, we propose a French-to-English cross-lingual summarization framework that analyzes the information in both languages to identify the most relevant sentences. In order to generate more informative cross-lingual summaries, we introduce the use of chunks and two compression methods at the sentence and multi-sentence levels. Experimental results on the MultiLing 2011 dataset show that our framework improves the results obtained by state-of-the art approaches according to ROUGE metrics.

Keywords: Cross-Language Automatic Text Summarization
Multi-Sentence Compression · Sentence Compression

1 Introduction

Cross-Language Automatic Text Summarization (CLATS) aims to generate a summary of a document where the summary language differs from the document language. The huge amount of information available on the Internet made it easier to be up to date on the news in the world. However, some information and viewpoints exist in languages that are unknown by readers. CLATS enables people who are not fluent in the source/target language to comprehend these data in a simple way.

The methods developed for CLATS can be classified, like the Automatic Text Summarization (ATS) domain, depending on whether they are extractive, compressive or abstractive [21]. The extractive ATS selects complete sentences that are supposed to be the most relevant of the documents; the compressive

This work was partially financed by the European Project CHISTERA-AMIS ANR-15-CHR2-0001.

M. Silberztein et al. (Eds.): NLDB 2018, LNCS 10859, pp. 467–479, 2018.
https://doi.org/10.1007/978-3-319-91947-8_48

ATS generates a summary by compression of sentences through the removal of non-relevant words; lastly, the abstractive ATS generates a summary with new sentences that are not necessarily contained in the original texts.

Many of the state-of-the-art methods for CLATS are of the extractive class. They mainly differ on how they compute sentence similarities and alleviate the risk that translation errors are introduced in the produced summary. Among these models, the CoRank method, which is characterized by its ability to simultaneously incorporate similarities between the original and translated sentences, turns out to be effective [22]. This method is extended in this paper in the following manner: we first take into account chunks instead of only words in the sentence similarity measures; then sentences are compressed in order to obtain a compressive CLATS system.

Inspired by the compressive ATS methods in monolingual analysis [1,5,10, 11,16,19,24], we adapt sentence and multi-sentence compression methods for the CLATS problem to just keep the main information. A Long Short Term Memory (LSTM) model is built to analyze a sentence and decide which words remain in the compression. We also use an Integer Linear Programming (ILP) formulation to compress similar sentences while analyzing both grammaticality and informativeness.

The remainder of this paper is organized as follows. In Sect. 2, we describe the most recent works about CLATS. Sections 3 presents our compressive CLATS approach. Section 4 reports the results achieved on the MultiLing 2011 dataset for the French-to-English task and shows that our method, particularly with the use of ILP for multi-sentence compression, outperforms the state of the art according to the ROUGE metrics. Finally, conclusions and future work are set out in Sect. 5.

2 Cross-Language Automatic Text Summarization

The first studies in cross-language document summarization analyzed the information in only one language [9,18]. Two typical CLATS schemes are the early and the late translations. The first scheme first translates the source documents to the target language, then it summarizes the translated documents using only information of the translated sentences. The late translation scheme does the reverse: it first summarizes the documents using abstractive, compressive or extractive methods, then it translates the summary to the target language.

Recent methods improved the quality of cross-lingual summarization using a translation quality score [2,23,25] and the information of the documents in both languages [22,26]. These methods are described in the next subsections.

2.1 Machine Translation Quality

Wan et al. trained a Support Vector Machine (SVM) regression method to predict the translation quality of a pair of English-Chinese sentences from basic features (such as sentence length, sub-sentence number, percentage of nouns and adjectives) and parse features (such as depth, number of noun phrases and verbal phrases in the parse tree) to generate English-to-Chinese CLATS [23]. They used 1,736 pairs of English-Chinese sentences (English sentences were translated automatically by Google Translate) and computed translation quality scores in a range from 1 to 5 (1 means "very bad" and 5 corresponds to "excellent"). The translation quality and informativeness scores were linearly combined to select the English sentences with both a high translation quality and a high informativeness:

$$score(s_i) = (1 - \lambda) \cdot InfoScore(s_i) + \lambda \cdot TransScore(s_i) \tag{1}$$

where $InfoScore(s_i)$ and $TransScore(s_i)$ are the informativeness score and translation quality prediction of the sentence s_i, and $\lambda \in [0, 1]$ is a parameter controlling the influence of the two factors. Finally, they translated the English summary to form the Chinese summary.

Similarly to Wan et al. [23], Boudin et al. used an ϵ-SVR to predict the translation quality score based on the automatic NIST metric as an indicator of quality [2]. They automatically translated English documents to French using Google Translate, then they analyzed some features (sentence length, number of punctuation marks, perplexities of source and target sentences using different language models, etc.) to estimate the translation quality of a sentence. They incorporated the translation quality score in the PageRank algorithm [3] to calculate the relevance of sentences based on the similarity between the sentences and the translation quality scores to perform English-to-French cross-lingual summarization (Eqs. 2–4).

$$p(V_i) = (1 - d) + d \times \sum_{V_j \in pred(V_i)} \frac{score(S_i, S_j)}{\sum_{V_k \in succ(V_i)} score(S_k, S_i)} p(V_i) \tag{2}$$

$$score(S_i, S_j) = Sim(S_i, S_j) \times Prediction(S_i) \tag{3}$$

$$Sim(S_i, S_j) = \frac{\sum_{w \in S_i, S_j} freq(w, S_i) + freq(w, S_j)}{\log(|S_i|) + \log(|S_j|)} \tag{4}$$

where d is the damping factor, $Prediction(s)$ is the translation quality score of the sentence s, $freq(w, s)$ is the frequency of the word w in the sentence s, $pred(V_i)$ and $succ(V_i)$ are the predecessors and successors vertices of the vertex V_i.

Inspired by the phrase-based translation models, Yao et al. proposed a phrase-based model to simultaneously perform sentence scoring, extraction and compression [25]. They designed a scoring scheme for the CLATS task based on a submodular term of compressed sentences and a bounded distortion penalty term. Their summary scoring measure was defined over a summary S as:

$$F(S) = \sum_{p \in S} \sum_{i=1}^{count(p,S)} d^{i-1} g(p) + \sum_{s \in S} bg(s) + \eta \sum_{s \in S} dist(y(s)) \tag{5}$$

where $g(p)$ is the score of phrase p (defined by the frequency of p in the document), $bg(s)$ is the bigram score of sentence s, $y(s)$ is the phrase-based derivation of the sentence s and $dist(y(s))$ is the distortion penalty term in the phrase-based translation models. Finally, d is a constant damping factor to penalize repeated occurrences of the same phrases, $count(p, S)$ is the number of occurrences of the phrase p in the summary S and η is the distortion parameter for penalizing the distance between neighboring phrases in the derivation.

2.2 Joint Analysis in both Languages

Wan proposed to leverage both the information in the source and in the target language for cross-lingual summarization [22]. In particular, he introduced two graph-based summarization methods (SimFusion and CoRank) for using both the English-side and Chinese-side information in the task of English-to-Chinese cross-lingual summarization. The first method linearly fuses the English-side and Chinese-side similarities for measuring Chinese sentence similarity. In a nutshell, this method adapts the PageRank algorithm to calculate the relevance of sentences, where the weight arcs are obtained by the linear combination of the cosine similarity of pairs of sentences for each language:

$$relevance(s_i^{cn}) = \mu \sum_{j \in D, j \neq i} relevance(s_j^{cn}) \cdot \tilde{C}_{ji}^{cn} + \frac{1-\mu}{n} \tag{6}$$

$$C_{ij}^{cn} = \lambda \cdot sim_{cosine}(s_i^{cn}, s_j^{cn}) + (1-\lambda) \cdot sim_{cosine}(s_i^{en}, s_j^{en}) \tag{7}$$

where s_i^{cn} and s_i^{en} represent the sentence i of a document D in Chinese and in English, respectively, μ is a damping factor, n is the number of sentences in the document and $\lambda \in [0, 1]$ is a parameter to control the relative contributions of the two similarity values. C^{cn} is normalized to $\tilde{C}^{cn}$ to make the sum of each row equal to 1. The CoRank method adopts a co-ranking algorithm to simultaneously rank both English and Chinese sentences by incorporating mutual influences between them. It considers a sentence as relevant if this sentence in both languages is heavily linked with other sentences in each language separately (source-source and target-target language similarities) and between languages (source-target language similarity) (Eqs. 8–12).

$$\mathbf{u} = \alpha \cdot (\tilde{\mathbf{M}}^{\mathbf{cn}})^T \mathbf{u} + \beta \cdot (\tilde{\mathbf{M}}^{\mathbf{encn}})^T \mathbf{v} \tag{8}$$

$$\mathbf{v} = \alpha \cdot (\tilde{\mathbf{M}}^{\mathbf{en}})^T \mathbf{v} + \beta \cdot (\tilde{\mathbf{M}}^{\mathbf{encn}})^T \mathbf{u} \tag{9}$$

$$M_{ij}^{\text{en}} = \begin{cases} \text{cosine}(s_i^{en}, s_j^{en}), & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$M_{ij}^{\text{cn}} = \begin{cases} \text{cosine}(s_i^{cn}, s_j^{cn}), & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\text{M}_{ij}^{\text{en,cn}} = \sqrt{\text{cosine}(s_i^{cn}, s_j^{cn}) \times \text{cosine}(s_i^{en}, s_j^{en})} \tag{12}$$

where M^{en} and M^{cn} are normalized to $\tilde{\text{M}}^{\text{en}}$ and $\tilde{\text{M}}^{\text{cn}}$, respectively, to make the sum of each row equal to 1. **u** and **v** denote the saliency scores of the Chinese and English sentences, respectively; α and β specify the relative contributions to the final saliency scores from the information in the same language and the information in the other language, with $\alpha + \beta = 1$.

Unlike Wan who generated extractive CLATS, Zhang et al. analyzed Predicate-Argument Structures (PAS) to obtain an abstractive English-to-Chinese CLATS [26]. They built a pool of bilingual concepts and facts represented by the bilingual elements of the source-side PAS and their target-side counterparts from the alignment between source texts and Google Translate translations. They used word alignment, lexical translation probability and 3-gram language model to measure the quality and the fluency of the Chinese translation, and the CoRank algorithm [22] to measure the relevance of the facts and concepts in both languages. Finally, summaries were produced by fusing bilingual PAS elements with an Integer Linear Programming (ILP) algorithm to maximize the saliency and the translation quality of the PAS elements.

3 Our Proposition

Following the CoRank-based approach proposed by Wan [22], we use his joint analysis of documents in both languages (source and target languages) to select the most relevant sentences. We expanded this method in three ways.

Firstly, we take into account Multi-Word Expressions (MWE) when computing similarities between sentences. These MWEs are very common in all languages and pose significant problems for every kind of NLP [20]. Their use in the context of CLATS helps the system to comprehend the semantic content of sentences. To realize a chunk-level tokenization, we used the Stanford CoreNLP tool for the English side [15]. This annotator tool, which integrates jMWE [8], detects various expressions, e.g., phrasal verbs ("*take off*"), proper names ("*San Francisco*"), compound nominals ("*cultivated plant*") or idioms ("*rain cats and dogs*"). Unfortunately, the tools developed for languages other than English have a lower coverage for MWEs. For this reason, MWEs were detected on the French side from the alignment of phrases inside parallel sentences using the Giza++ application [17].

A second evolution of the CoRank-based approach is the use of a Multi-Sentence Compression (MSC) method to generate more informative compressed outputs from similar sentences. For this purpose, the sentences are grouped in clusters based on their similarity in both languages. For each cluster with more than one sentence, which is common in the case of multi-document summarization, a MSC method guided by keywords is applied to build a sentence with the core information of the cluster [13,14].

A third extension of the approach relies on compression techniques of a single sentence by deletion of words [5]. Still with the idea to generate more informative summaries, sentence compression is applied for sentences that stand alone during the clustering step required by the MSC step.

The following subsections describe in detail the architecture of our system.

3.1 Preprocessing

Initially, French texts are translated to English using the Google Translate system, which is at the cutting edge of the statistical translation technology and was used in the majority of the state-of-the-art CLATS methods.

Then, chunks are identified inside the English texts with the Stanford CoreNLP, while the English and French parallel sentences are aligned with the Giza++ toolkit.[1] Two Giza++ models were trained on the Europarl v7 (2.1 M sentence pairs[2]) and News-Commentary 11 (0.2 M sentence pairs[3]) datasets in both directions (English-to-French and French-to-English). Like the training corpora used for statistical translation models, the alignments obtained by both models were intersected by the default heuristic *grow-diag-final* of the Moses toolkit [7]. From these alignments and the English MWEs, a chunk-level tokenization is performed on the French side.

Finally, sentences are clustered according to their similarities, sentences with a similarity score bigger than threshold θ remain in the same group. The similarity score of a pair of sentences i and j is defined by the cosine similarity in both languages:

$$sim(i,j) = \sqrt{\text{cosine}(s_i^{fr}, s_j^{fr}) \times \text{cosine}(s_i^{en}, s_j^{en})} \tag{13}$$

where s_i^{fr} and s_i^{en} represent a sentence i in the French and English languages.

3.2 Sentence and Multi-Sentence Compression

To avoid the accumulation of errors that would appear in a translation-compression-translation pipeline, we restrict the sentence and multi-sentence compressions to the sentences in the target language.

Sentence Compression. The Sentence Compression (SC) problem is here seen as the task to delete non-relevant words in a sentence [5,10,11,19,24]. Filippova et al. [5] used an LSTM model to compress sentences by deletion of words. In few words, this model follows a sequence-to-sequence paradigm to verify which words of a sentence c remain in the compression. A word i in a sentence c is represented by its word embedding and the word embedding of its parent node in the parse tree. Then, a first LSTM encodes this sentence and another LSTM generates the sequence of the words that are kept in the compression. LSTMs are composed of input i_t, control state c_t and memory state m_t that are updated at time step t (Eqs. 14–19).

$$i_t = \text{sigm}(W_1 x_t + W_2 h_{t-1}) \tag{14}$$

$$i'_t = \tanh(W_3 x_t + W_4 h_{t-1}) \tag{15}$$

[1] The GIZA++ model, https://github.com/moses-smt/giza-pp.

[2] http://www.statmt.org/europarl/.

[3] http://opus.nlpl.eu/News-Commentary.php.

$$f_t = \text{sigm}(W_5 x_t + W_6 h_{t-1}) \tag{16}$$

$$o_t = \text{sigm}(W_7 x_t + W_8 h_{t-1}) \tag{17}$$

$$m_t = m_{t-1} \odot f_t + i_t \odot i'_t \tag{18}$$

$$h_t = m_t \odot o_t \tag{19}$$

where the operator $\odot$ denotes element-wise multiplication, the matrices $W_1, ...,$ W_8 and the vector h_0 are the parameters of the model, and all the non-linearities are computed element-wise (more details in [5]). Contrary to [5], we analyze the sentence at the chunk level, so we remove a chunk only if all words of this chunk were deleted in the SC process described above.

Multi-Sentence Compression. For the clusters that have more than a sentence, we use a Chunk Graph (CG) to represent them and an ILP method to compress these sentences in a single, short, and hopefully correct and informative sentence. Among several state-of-the-art MSC methods [1,4,16], Linhares Pontes et al. [13,14] used an ILP formulation to guide the MSC using a list of keywords. Our system incorporates this approach to create a Word Graph and to calculate the weight arcs (cohesion between the words, Eqs. 20 and 21), but instead of restricting to single words we also consider multi-word chunks (Chunk Graph):

$$w(i,j) = \frac{\text{cohesion}(i,j)}{\text{freq}(i) \times \text{freq}(j)}, \tag{20}$$

$$\text{cohesion}(i,j) = \frac{\text{freq}(i) + \text{freq}(j)}{\sum_{s \in C} \text{diff}(s,i,j)^{-1}}, \tag{21}$$

where freq(i) is the chunk frequency mapped to the vertex i and the function diff(s,i,j) refers to the distance between the offset positions of chunks i and j in the sentences s of a cluster C containing these two chunks. From the relevance of the 2-grams[4] (Eq. 20), we consider that the relevance of a 3-gram is based on the relevance of their two inner 2-grams, as described in Eq. 22:

$$\text{3-gram}(i,j,k) = \frac{qt_3(i,j,k)}{\max_{a,b,c \in CG} qt_3(a,b,c)} \times \frac{w(i,j) + w(j,k)}{2}, \tag{22}$$

where $qt_3(i,j,k)$ is the number of 3-grams composed of chunks in i, j and k vertices in the cluster. The 3-grams increase the grammatical quality of the compression.

We also use Latent Dirichlet Allocation (LDA) to identify the keywords at the global (all texts of a topic) and local (cluster of similar sentences) levels to have the gist of a document and of a cluster of similar sentences. Then, an ILP method, as described in [13,14], generates a compression guided by keywords, in order to favor informativeness and grammaticality as expressed in Eq. 23.

[4] In this work, a unigram is represented by a chunk.

In other words, this method looks for a path (sentence) that has a good cohesion and contains a maximum of keywords.

$$\text{minimize}\ \Big(\sum_{(i,j)\in A} w(i,j)\cdot x_{i,j} - c\cdot\sum_{k\in K} b_k - \sum_{t\in T} d_k\cdot z_t\Big) \tag{23}$$

where x_{ij} indicates the existence of the arc (i,j) in the solution, $w(i,j)$ is the cohesion of the chunks i and j (Eq. 20), K is the set of labels (each representing a keyword), b_k indicates the existence of a chunk with a keyword k in the solution, c is the keyword bonus of the graph,[5] T is the set of 3-grams in the cluster, d_t indicates the existence of the 3-gram t in the solution and z_t represents the relevance of the 3-gram t defined by the Eq. 22. Finally, we generate the 50 best solutions according to the objective (23) and we select the compression with the lowest normalized score (Eq. 24) as the best compression.

$$\text{score}_{norm}(s) = \frac{e^{\text{score}_{opt}(s)}}{||s||}, \tag{24}$$

where $\text{score}_{opt}(s)$ is the score of the sentence s from Eq. 23.

3.3 CoRank Method

The CoRank method adopts a co-ranking algorithm to simultaneously rank both French and English sentences by incorporating mutual influences between them. We use the CoRank method (Sect. 2.2) to calculate the relevance of sentences. In order to avoid the accumulation of errors that would be generated by a translation-compression-translation pipeline, similarity is computed from the uncompressed versions of sentences and that is only in the last summary generation step that compressed sentences are used.

Finally, as usual for ATS, a summary is generated with the most relevant sentences and the sentences redundant with the ones that have already been selected are put aside.

4 Experimental Results

In order to analyze the performance of our method, we compare it with the early translation, the late translation, the SimFusion and the CoRank methods [22]. The early and late translations are based on the SimFusion method, the differences between the systems being on the similarity metric (Eq. 7) computed either in the target language (early translation) or in the source language (late translation). We analyzed three versions of SimFusion with $\lambda = 0.25$, 0.50 and 0.75. The CoRank method uses $\alpha = \beta = 0.5$. We generated three versions of

[5] The keyword bonus allows the generation of longer compressions that may be more informative and it is defined by the geometric average of all weight arcs in the Chunk Graph.

our approach, named Compressive CLATS (CCLATS): SC, MSC and SC+MSC. The first version uses the SC method to compress sentences, the MSC method compresses clusters of similar sentences and extracts the rest of the sentences, and the last version applies MSC to clusters of similar sentences and SC to other sentences.

We compress only sentences with more than 15 words and we preserve compressions with more than 10 words to avoid short outputs with little information. The MSC method selects the 10 most relevant keywords per topic and the 3 most relevant keywords per cluster of similar sentences to guide the compression generation. All systems generate summaries composed of 250 words with the most relevant sentences, while the redundant sentences are discarded. We apply the cosine similarity measure with a threshold θ of 0.5 to create clusters of similar sentences for the MSC and to remove redundant sentences in the summary generation.

We use the pre-trained word embeddings[6] with 300-dimensional embeddings and an LSTM model with only one layer with 256-dimensional embeddings. Our Neural Network is trained on the publicly released set of 10,000 sentence-compression pairs.[7]

4.1 Dataset

We used the MultiLing Pilot 2011 dataset [6] derived from publicly available WikiNews English texts. This dataset is composed of 10 topics, each topic having 10 source texts and 3 reference summaries. Each reference summary contains a maximum of 250 words. Native speakers translated this dataset into Arabic, Czech, French, Greek, Hebrew and Hindi languages. Specifically, we use English and French texts to test our system.

4.2 Automatic Evaluation

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 [12]. In particular, we used the f-measure metrics ROUGE-1, ROUGE-2 and ROUGE-SU4.

Table 1 shows the ROUGE f-measure scores achieved by each system using the MultiLing Pilot 2011 dataset. The baselines, especially the late translation, have the worst scores. Similarly to the results described in [22], the CoRank method outperforms the SimFusion method. The analysis of the output of the CCLATS versions brought to light that the SC version removed relevant information of sentences, achieving lower ROUGE scores than CoRank. CCLATS.MSC generated more informative summaries and leads to the best ROUGE scores. Finally, the SC+MSC version obtains better results than other systems but still does not reach the highest ROUGE scores measured when using MSC alone.

[6] Publicly available at: code.google.com/p/word2vec.

[7] http://storage.googleapis.com/sentencecomp/compression-data.json.

Table 1. ROUGE f-measure scores for the French-to-English CLATS using the MultiLing Pilot 2011 dataset. * indicates the results are statistically better than baselines and the SimFusion method with a 0.05 level.

Methods	ROUGE-1	ROUGE-2	ROUGE-SU4
baseline.early	0.41461	0.10251	0.16001
baseline.late	0.41137	0.10270	0.15795
SimFusion.$\lambda = 0.25$	0.41403	0.10545	0.16081
SimFusion.$\lambda = 0.50$	0.41198	0.10268	0.15820
SimFusion.$\lambda = 0.75$	0.41516	0.10397	0.15992
CoRank	0.45552	0.12952	0.19056
CCLATS.SC	0.45436	0.11809	0.18463
CCLATS.MSC	**0.47221***	**0.13613**	**0.19881***
CCLATS.SC+MSC	0.46786*	0.13056	0.19420*

4.3 Discussion

The lower results of the early and late translations with respect to other systems prove that the texts in each language provide complementary information. It also establishes that the analysis of sentences in the target language plays a more important place to generate informative cross-lingual summaries. As seen for English-to-Chinese CLATS [22], the CoRank method generates better results than the baselines and SimFusion because it considers the information in each language separately and together, while the baselines restrict the analysis of sentence similarity to one language separately and the SimFusion method analyzes only the cross-lingual sentence similarity.

It is expected that a piece of information found in several texts is relevant for a topic. In accordance with this principle, the MSC method looks for the repeated information and generates a short compression with selected keywords that summarize the main information. The two kinds of keywords (global and local) guide the MSC method to generate compression linked to the main topic of the documents and to the specific information presented in the cluster.

With regard to SC, this compression method did not produce good results in our experiments. This observation may be explained by the reduced size of the corpus we used to train our NN (10,000 parallel sentence-compression instance), while the system described in Filippova et al. [5] could benefit from a corpus of about two million instances. Whereas the CCLATS.MSC version leaves unchanged the sentences that do not have similar sentences, the SC+MSC version involves the SC model to compress these sentences. As the CCLATS.SC system has lower performance than the pure extractive CoRank method, the SC+MSC also had lower results than MSC version.

A difference between the SC and MSC approaches is that MSC uses global and local keywords to guide the compression preserving the main information, while the SC method does not realize this kind of analysis. Another difference between them is that the MSC method does not need a training corpus to generate compressions.

To sum up, the joint analysis of both languages with CoRank helps the generation of cross-lingual summaries. On the one hand, the SC model deletes relevant information, thereby reducing the informativeness of summaries. On the other hand, the MSC method proves to be a good alternative to compress redundant information and to preserve relevant information. Finally, the CCLATS.MSC greatly improves the ROUGE scores and significantly outperforms the baselines and the SimFusion methods.

5 Conclusion

In this paper we have proposed two compressive methods to improve the generation of cross-lingual summaries. The proposed system analyzes a document in both languages to extract all of the relevant information. Then, it applies two kinds of methods to compress sentences. Unlike the sentence compression system (CCLATS.SC) that needs a large training dataset to generate compressions of good quality, the multi-sentence compression version of our system (CCLATS.MSC) generates better ROUGE results than extractive Cross-Language Automatic Text Summarization systems. Moreover, it has the advantage of not requiring a training corpus to generate summaries of good quality.

There are several avenues worth exploring from this work. First, we want to investigate how the size of the training data of our Neural Network to generate sentence compressions acts upon the quality of the summaries. It would also be interesting to include an attention mechanism in our Neural Network to analyze the sentence and the gist of the topic. Finally, our evaluation was confined to ROUGE scores, which mostly measure the informativeness. An additional human evaluation must be performed to confirm that the informativeness and the grammaticality are improved with the use of compression methods.

References

1. Banerjee, S., Mitra, P., Sugiyama, K.: Multi-document Abstractive Summarization Using ILP Based Multi-sentence Compression. In: 24th International Conference on Artificial Intelligence (IJCAI), IJCAI 2015, pp. 1208–1214 (2015)
2. Boudin, F., Huet, S., Torres-Moreno, J.: A graph-based approach to cross-language multi-document summarization. Polibits **43**, 113–118 (2011)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**(1–7), 107–117 (1998)
4. Filippova, K.: Multi-sentence compression: finding shortest paths in word graphs. In: COLING, pp. 322–330 (2010)
5. Filippova, K., Alfonseca, E., Colmenares, C.A., Kaiser, L., Vinyals, O.: Sentence compression by deletion with LSTMs. In: EMNLP, pp. 360–368 (2015)
6. Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., Varma, V.: TAC2011 multiling pilot overview. In: 4th Text Analysis Conference TAC (2011)

7. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: 45th Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume, pp. 177–180 (2007)
8. Kulkarni, N., Finlayson, M.A.: jMWE: a Java toolkit for detecting multi-word expressions. In: Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE), pp. 122–124 (2011)
9. Leuski, A., Lin, C.Y., Zhou, L., Germann, U., Och, F.J., Hovy, E.: Cross-lingual C*ST*RD: English access to Hindi Information. J. ACM Trans. Asian Lang. Inf. Process. **2**(3), 245–269 (2003)
10. Li, C., Liu, F., Weng, F., Liu, Y.: Document summarization via guided sentence compression. In: EMNLP, pp. 490–500. ACL (2013)
11. Li, C., Liu, Y., Liu, F., Zhao, L., Weng, F.: Improving multi-documents summarization by sentence compression based on expanded constituent parse trees. In: EMNLP, pp. 691–701. ACL (2014)
12. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Workshop Text Summarization Branches Out (ACL 2004), pp. 74–81 (2004)
13. Linhares Pontes, E., Huet, S., Gouveia da Silva, T., Linhares, A.C., Torres-Moreno, J.M.: Multi-sentence compression with word vertex-labeled graphs and integer linear programming. In: Proceedings of TextGraphs-12: the Workshop on Graph-based Methods for Natural Language Processing. Association for Computational Linguistics (2018)
14. Linhares Pontes, E., Gouveia da Silva, T., Linhares, A.C., Torres-Moreno, J.M., Huet, S.: Métodos de otimização combinatória aplicados ao problema de compressão multifrases. In: Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), pp. 2278–2289 (2016)
15. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations, pp. 55–60 (2014)
16. Niu, J., Chen, H., Zhao, Q., Su, L., Atiquzzaman, M.: Multi-document abstractive summarization using chunk-graph and recurrent neural network. In: IEEE International Conference on Communications, ICC, pp. 1–6 (2017)
17. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**(1), 19–51 (2003)
18. Orasan, C., Chiorean, O.A.: Evaluation of a cross-lingual Romanian-English multi-document summariser. In: 6th International Conference on Language Resources and Evaluation (LREC) (2008)
19. Qian, X., Liu, Y.: Fast joint compression and summarization via graph Cuts. In: EMNLP, pp. 1492–1502 (2013)
20. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword expressions: a pain in the neck for NLP. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 1–15. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45715-1_1
21. Torres-Moreno, J.M.: Automatic Text Summarization. Wiley and Sons, London (2014)
22. Wan, X.: Using bilingual information for cross-language document summarization. In: ACL, pp. 1546–1555 (2011)
23. Wan, X., Li, H., Xiao, J.: Cross-language document summarization based on machine translation quality prediction. In: ACL, pp. 917–926 (2010)

24. Yao, J., Wan, X., Xiao, J.: Compressive document summarization via sparse optimization. In: IJCAI, pp. 1376–1382. AAAI Press (2015)
25. Yao, J., Wan, X., Xiao, J.: Phrase-based compressive cross-language summarization. In: EMNLP, pp. 118–127 (2015)
26. Zhang, J., Zhou, Y., Zong, C.: Abstractive cross-language summarization via translation model enhanced predicate argument structure fusing. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(10), 1842–1853 (2016)