Automating News Summarization with Sentence Vectors Offset

Mauricio Steinert*, Roger Granada*, João Paulo Aires* and Felipe Meneguzzi†
Graduate Program in Computer Science, School of Technology
Pontifical Catholic University of Rio Grande do Sul (PUCRS), Brazil
Email: *{mauricio.steinert, roger.granada, joao.aires.001}@acad.pucrs.br, †felipe.meneguzzi@pucrs.br

***Abstract*—Text summaries consist of short versions of texts that convey their key aspects and help readers understand the gist of such texts without reading them in full. Generating such summaries is important for users who must sift through ever increasing volumes of content generated on the web. However, generating high-quality summaries is time consuming for humans and challenging for automated systems, since it involves understanding the semantics of the underlying texts in order to extract key information. In this work, we develop an extractive text summarization method using vector offsets, which we show empirically to be able to summarize texts from an Internet news corpus with an effectiveness competitive with state-of-the-art extractive techniques.**

***Index Terms*—automatic text summarization, natural language processing, information retrieval, word embedding.**

I. Introduction

In recent years, there has been an increasing availability of information from a variety of sources, mainly in the World Wide Web, which contains billions of documents and is exponentially growing. Such availability makes people face an overload of information. In order to alleviate the content overload, automatic summarization systems intend to produce a concise and fluent summary of the most important information. Thus, automatic text summarization produces a summary, *i.e.*, a short length version of the source text that contains their most relevant content. Text summarization is a challenging task that must deal with issues such as redundancy, temporal dimension and other aspects related to text writing [1].

The first automatic text summarization method dates from 1950 [2] motivated by the growing number of academic articles available at that time. This problem is exacerbated nowadays since the Internet allows us to access massive amounts of textual information, and we need an efficient way to prioritize reading material. Automating summarization of such information is useful because it allows users to quickly navigate over all this information. Manually generating summaries is a time consuming task that depends on writers subjectivity and personal opinions on the matter [2], being infeasible due to the amount of information we have available nowadays.

The text summarization task is broadly classified as *extractive summarization*, where snippets of text like words, sentences or paragraphs that better represent texts whole content are selected as summary, and *abstractive summarization*, where summaries are generated by paraphrasing key concepts in text, avoiding reusing material from original text [3]. In this paper, we develop an automatic text summarization method based on word vector offset, which operates by comparing similarity between a vector representation of whole text with each sentence in the text. Traditional text summarization techniques are based on term-frequency and other statistical features to identify relevant words in text. By contrast, modern approaches use learning-based techniques that requires large volumes of labeled data to train a neural network solution. Unlike previous work that perform extractive summarization using Recurrent Neural Networks with attention mechanisms, in this paper, we use a simpler method based on the offset of vectors representing sentence embeddings. Our model is capable of providing reasonable results without a training stage for summarization task, yielding effective results based on a simple implementation.

II. Background

A. Word and Sentence Embedding

In the last years, much attention has been given to word embeddings (Word2Vec) since they can map words into a low dimensional space to capture their lexical and semantic properties [4]–[7]. In order to perform this mapping between raw words and low dimensional spaces, Word2Vec uses internal representations of neural network models, such as feed-forward models [8], recurrent neural network (RNN) models [4], or by low-rank approximation of co-occurrence statistics [6].

One of the main algorithms to create embeddings of words is called Continuous Bag-of-Words (CBOW) [5]. CBOW is based on the Harris' [9] assumption which says that words that occur in similar contexts have similar meaning. Thus, each word in CBOW is represented by the context where it occurs, *i.e.*, the generated representation of a target word is given by its neighboring words using a window with a defined size. To do so, CBOW employs a neural network composed of three layers (input, projection, and output layers), where the input layer receives a sequence of one-hot encoding vectors representing context words. In this configuration, the CBOW model uses n words from the history and n words from the future as context words, where all these context words get projected into the same position, *i.e.*, the target word is represented as the average of its contexts. Given the context words in a window,

the training criterion is to correctly classify the target word – the word at the center of the window [5].

Sentence embedding (Sent2Vec) is an extension of word embeddings (Word2Vec) [5] that learns the representation of sentences instead of words. Pagliardini *et al.* [10] affirm that Sent2Vec can be interpreted as a natural extension of the word contexts from CBOW to a larger sentence context, with the sentence words being specifically optimized towards additive combination over the sentence, by means of the unsupervised objective function. Thus, the sentence embedding v_S is represented as the average of the embeddings of its constituent words (v_w) as presented in Equation 1, where $R(S)$ is the list of n-grams (including unigrams) present in sentence S.

$$v_S := \frac{1}{|R(S)|} \sum_{w \in R(S)} v_w \tag{1}$$

As performed by Mikolov *et al.* [4] in Word2Vec, Sent2Vec performs random sub-sampling by deleting random words once all the n-grams have been extracted in order to improve generality. Missing words are predicted from the context by using a softmax output approximated by negative sampling.

B. Vector Offset

Word embeddings are surprisingly good at capturing syntactic and semantic regularities in language [11], [12], *i.e.*, they are able to capture the relationships between words in an expressive way. Such regularities are observed as constant vector offsets between pairs of words sharing a particular relationship, *e.g.*, we can observe that the subtraction of two vectors ($v_{\text{apple}} - v_{\text{apples}}$) generates a vector encoding the meaning of *singular/plural*. The subtraction of two other vectors with the same relation (*e.g.*, $v_{\text{car}} - v_{\text{cars}}$) generates a similar embedding vector. In this sense, all pairs of words that share a relation are related by the same offset in the embedding space.

These semantic regularities allow us to perform vector operations on embeddings for finding similar vectors, such as finding the vector representing the word *Queen* by using $v_{\text{Queen}} \approx (v_{\text{King}} - v_{\text{Man}}) + v_{\text{Woman}}$. Hence, given two pairs of words that share the same semantic relation $v_a : v_a^*, v_b : v_b^*$, the relation between those two words can be represented as

$$v_a^* - v_a \approx v_b^* - v_b \tag{2}$$

In this study, we assume that the linguistic regularities captured by word embeddings are extended to sentence embeddings.

C. ROUGE Scores

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [13] is a metric used for evaluating automatic summarization, *i.e.*, for automatically determining the quality of a summary. It compares a given summary to other (ideal) summaries created by humans by counting the number of overlapping units. These units may vary according to the type of ROUGE metric, *e.g.*, measuring the overlapping between either unigrams, bigrams, or n-grams between the given summary and the ideal summary. In this work, we use ROUGE-N and ROUGE-L scores, where N represents the n-gram used to compare sentences, *e.g.*, ROUGE-1 for unigrams, ROUGE-2 for bigrams, and so on. ROUGE-L uses the longest common sub-sequences (LCS) to compute the similarity between sentences.

III. Vector Offset Summarization

Inspired by Aires *et al.* [7], which use norm embedding offset to identify norm conflict in contracts, we develop the Vector Offset Summarization (VOS) as an extractive summarization method. Our method ranks sentences that best represent the overall content of documents based on vector offset of the embedding representing the mean of the whole document and the embedding of each sentence in the document.

In order to do so, we first process documents in order to extract their sentences and the abstractive summary, *i.e.*, ground truth sentences are marked in the text with the *@highlight* tag. We convert each sentence into an embedding vector representation using Sent2vec [10]. Next, we calculate the mean vector representation (v_{mean}) (Equation 3) using all sentence embeddings of a document, where $\mathcal{D}$ is a set with all sentence embeddings of a document, and v_s represents each sentence vector in document.

$$v_{\text{mean}} = \frac{1}{|\mathcal{D}|} \sum_{v_s \in \mathcal{D}} v_s \tag{3}$$

Having a vector representing the mean of the sentence embeddings, we can generate the offset vector by subtracting it by the vector representing the ground truth vector, *i.e.*, the *@highlight* vector. As *@highlight* represents a summary of the document and the mean of the document represents the semantic of its content, when subtracting the summary from the mean of the document, the offset concentrates the semantic of a summary regardless the content of the document. For example, consider we have a document describing sport activities. The mean of its sentence embeddings represents the semantic of sport activities. On the other hand, the *@highlight* is composed by the summary plus the content, which is the sport activities. When subtracting the *@highlight* from the mean, what remains is only the semantic content representing a general summary. The offset vector is calculated according to Equation 4, where $v_{\text{@highlight}}$ represents the embedding vector generated from sentences marked as *@highlight*.

$$v_{\text{offset}} = v_{\text{mean}} - v_{\text{@highlight}} \tag{4}$$

In order to decide if a sentence must belong to a summary, we add to it the sense of the summary produced by the offset vector and compute the distance to the vector representing the mean using the Frobenius norm. Equation 5 defines this distance, where v_{sc} is the vector representing the summary candidate.

$$\begin{aligned} v_{sc} &= v_s + v_{\text{offset}} \\ d &= ||v_{sc} - v_{\text{mean}}||_F \end{aligned} \tag{5}$$

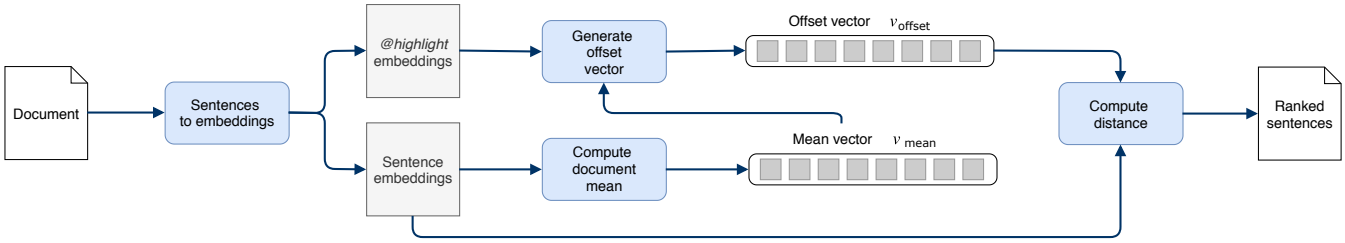

Fig. 1. Pipeline of the Vector Offset Summarization method.

After calculating the distance to the mean (Equation 5) for each sentence of the document, we ranked them according to their distance (d). The summary is selected based on candidate sentences that have the lowest distance values until a specific summary length is reached. Figure 1 illustrates the pipeline of our method. We make the code and evaluation results available in the project's repository[1].

IV. Experiment

In this section we describe the dataset we used to perform experiments and its pre-processing, the gold standard creation, the embedding vectors, and how we evaluate our experiments. The pre-processing generates a subset of the dataset by discarding small sentences and short stories. Gold standard creation is necessary since the dataset contains human written abstractive summaries and not extractive summaries. Embedding vectors are used to convert sentences from raw text files into their embedding representations. Finally, evaluation is performed in terms of Precision, Recall and F-score using ROUGE scores.

A. Dataset

We evaluate VOS using CNN/Daily Mail dataset[2] [14]. This dataset contains stories from CNN and Daily Mail websites and human generated abstractive summary from these stories (marked as *@highlight*). It was originally developed for automating question and answer tasks, but it has been recently used in text summarization tasks [15]–[17]. We select this dataset since its size allows us to compare with the state-of-the-art algorithms, such as Recurrent Neural Networks. The dataset contains a total of 182,750,863 words spread over 6,032,961 sentences in 312,085 documents (stories). The average size of each story is 833 words within 27 sentences.

In preliminary experiments, we observed the negative impact caused by overly short sentences and stories. We conjecture that short sentences do not contain enough semantic information to create consistent embeddings, as well as overly short stories already represent a summarized version of the news. Hence, we define constraints to guarantee that our summaries generate consistent embeddings across stories. Our constraints include discarding sentences shorter than 30 characters or larger than 5,000 characters, and stories shorter than 12 sentences. Under these constraints, we evaluate our results over 297,389 documents (stories). Finally, we convert all characters to lower case to match the dictionaries used to train the word embedding, which only contain lower case words.

In order to convert sentences from raw text files into embedding vectors we use the Sent2vec library[3]. The Sent2Vec authors make available a series of pre-trained models from different sources, among them there is an unigram and a bigram models trained on the English Wikipedia corpus [10].

We used the unigram pre-trained model composed by 1,066,988 words that generates embeddings such that each word yields an embedding consisting of 600 floating point values [10] (hereafter called *VOS-600* model). We trained a second model containing 2,809,163 words that yields embeddings of 100 floating point values per word (hereafter called *VOS-100*). Both models were trained using the English Wikipedia corpus.

B. Gold Standard Creation and Evaluation

Since we perform extractive summarization, we have to identify relevant sentences inside the text to compose the summary. However, the dataset contains human written abstractive summaries as ground truth, *i.e.*, the ground truth is composed of sentences that are not themselves in the document. These ground truth sentences are marked in the text with the *@highlight* tag. Using these highlights, we have to find in the text sentences that are representative enough to be the summary.

Nallapati *et al.* [16] creates new ground truth based on an unsupervised approach to convert abstractive summaries to extractive labels, and uses these labels as input to train a neural network, where the most representative sentences are the ones that maximize ROUGE scores with respect to ground truth sentences of the abstractive summaries. Similar to Nallapati *et al.*, our gold standard contains the top 10 sentences ranked according their ROUGE score for each story.

We evaluate our method in terms of Precision, Recall and F-score of the ROUGE-N and ROUGE-L scores. For ROUGE-N, we use unigrams (ROUGE-1) and bigrams (ROUGE-2). We compute Precision as $\mathcal{P} = \frac{\mathcal{T}_s \cap \mathcal{T}_g}{\mathcal{T}_s}$, Recall as $\mathcal{R} = \frac{\mathcal{T}_s \cap \mathcal{T}_g}{\mathcal{T}_g}$ and F-score as $\mathcal{F} = \frac{2 \cdot \mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}}$, where $\mathcal{T}_s$ represents the terms

[1]https://github.com/mauriciosteinert/text-summarization-offset

[2]https://cs.nyu.edu/~kcho/DMQA/

[3]https://github.com/epfml/sent2vec

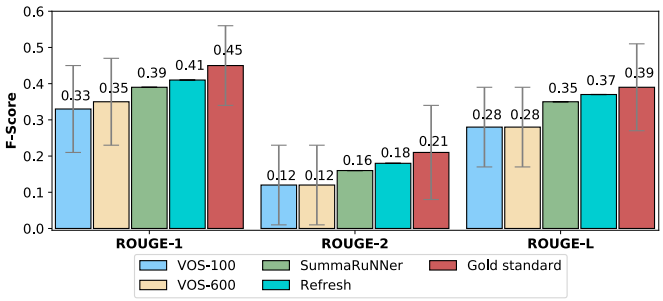

Fig. 2. Performance evaluation of vector offset method using ROUGE F-scores.

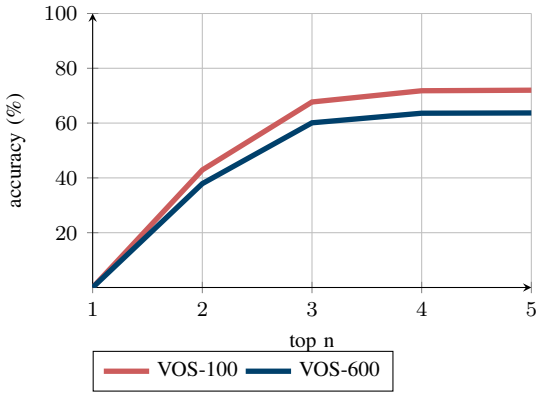

Fig. 3. Summarization accuracy given top 5 sentences according to ROUGE scores.

(n-grams) in the story being processed and $\mathcal{T}_g$ represents the n-grams in the gold standard.

V. Results

We ranked all summaries using the offset of the 100-dimension sentence vectors (VOS-100) and the offset of the 600-dimension sentence vectors (VOS-600), and compare to the summaries generated by selecting sentences with the highest ROUGE scores (Gold Standard). We use ROUGE-N scores, where ROUGE-1 stands for unigrams and ROUGE-2 bigrams, and ROUGE-L for the longest common subsequence. Table I shows precision ($\mathcal{P}$), recall ($\mathcal{R}$) and F-score ($\mathcal{F}$) for our Vector Offset Summarization (VOS) methods and for the gold standard (Gold std). In this context, gold standard means the highest ROUGE score a method can achieve.

TABLE I
Detailed scores evaluation for VOS-100, VOS-600 and Gold Standard.

Experiment	ROUGE-1			ROUGE-2			ROUGE-L		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
VOS-100	0.41	0.31	0.34	0.15	0.11	0.12	0.36	0.27	0.28
VOS-600	0.42	0.32	0.35	0.17	0.12	0.13	0.38	0.28	0.29
Gold Std	**0.52**	**0.43**	**0.45**	**0.25**	**0.20**	**0.21**	**0.48**	**0.39**	**0.39**

Comparing our both approaches, we observe that VOS-600 yields better results when compared to VOS-100. Although the results appear related to the size of the embedding, *i.e.*, the larger the better, they are not proportional to the vector dimensionality.

Figure 2 illustrates the F-score achieved at each ROUGE-N and ROUGE-L measure, where error bars report standard deviation computed over dataset examples. As we can see, our method does not affect significantly standard deviation values when compared to the gold standard. We compare our results with SummaRuNNer [16] and REFRESH [18] methods running over the same dataset. When analysing the results, we observe that our method gets close results compared to these, but using a much simpler algorithm.

Figure 3 illustrates the percentage of documents where our VOS methods select the sentences ranked in top 5 best sentences according to the ROUGE score evaluation. Observing the results, 72% of the sentences that VOS-600 selected are in the top 5 sentences classified by the ROUGE score, whereas 63% of the sentences are selected by VOS-100.

A. Qualitative analysis

ROUGE metrics are useful for automating sentences similarity over large datasets. On the other hand, as it is based on n-gram models that evaluate precise match between sentence elements, it is not clear how good a summary is. Hence, in this Section we manually evaluate some generated summaries from a subjective point of view, taking into account text conciseness and completeness when compared to source text and ground truth summary.

Table II shows our first example, which is an announcement about upcoming movies. The Gold standard summary emphasizes four major pieces of information: first about Paul Bettanys' character Vision, second about actor Charlie Cox as the Daredevil character, third The Thing character in Fantastic Four franchise, and fourth, the first look at the Angel character in "X-Men: Apocalypse" movie. Gold standard based on ROUGE score selects the sentence that references "X-Men" and "Fantastic Four" movies. VOS-600 instead selects the sentence that references the Vision character, which by evaluating source document are the predominant manner in text that have three paragraphs talking about it. VOS-600 second sentence talks about "X-Men" movie, which shares similar text space when compared to other remaining movies.

Table III shows our second example about how the greenhouse effect and other climate changes affect the life in Shishmaref, Alaska. In this story, there are no common sentences selected by VOS-600 and Gold standard. As expected, the best ROUGE scores are attributed to sentences related to what is established by ground truth summary. On the other hand, VOS-600 selects sentences based on what is most frequent in the text, *i.e.*, what is the situation of people that live in this area and have to relocate due to climate changes.

VI. Related Work

Automating text summarization first attempt dates from 1950 and uses statistical information like term frequency to

TABLE II
QUALITATIVE EXAMPLE 2.

Source	Summary	ROUGE 1	2	L
Ground truth	marvel studios releases first looks at paul bettany as the vision in "avengers: age of ultron" and charlie cox in full "daredevil" costume. jamie bell's character of the thing was also unveiled for 20th century fox's marvel-based reboot of "fantastic four". bryan singer unveiled the first look at "x-men: apocalypse" angel played by ben hardy.	-	-	-
VOS-600	with less than a month to go before the movie hits theaters, marvel studios put all the speculation to rest with a poster featuring bettany as the heroic android, who was a member of the superhero group for many years in the comics. not to be outdone, director bryan singer announced a new character for next year's sequel "x-men: apocalypse," by telling empire magazine that ben hardy would be playing the role of the winged mutant angel.	32	9	28
Gold standard	not to be outdone, director bryan singer announced a new character for next year's sequel "x-men: apocalypse," by telling empire magazine that ben hardy would be playing the role of the winged mutant angel. and thursday's new super images weren't quite done, because the questions over how jamie bell's rocky character the thing in the rebooted "fantastic four" movie (out august 7) might look were also finally answered.	38	10	30

TABLE III
QUALITATIVE EXAMPLE 3.

Source	Summary	ROUGE 1	2	L
Ground truth	u.n. panel releases the first part of its new climate assessment this week. john sutter: the impact of climate change is obvious, everywhere. sutter says lawmakers should look to alaska for evidence of the effects. villages there are thinking of relocating because of changes in the climate.	-	-	-
VOS-600	another community, newtok, which was the subject of a fascinating series by the guardian, is actually in the process of relocating now, according to sexauer. the tiny inupiat eskimo community in near-arctic alaska – which i was lucky enough to visit on a reporting trip in 2009 and which is home to some of the sweetest and most colorful people you'll meet – has been watching climate change happen to it for years now.	23	7	16
Golden standard	intergovernmental panel on climate change continues to update all of us on the latest science and evidence. but it's already obvious to everyone paying attention that we need to act in new and profoundly urgent ways to blunt the future impact of climate change, and to mitigate the changes that are already taking shape all over the world.	31	6	21

identify key aspects of text [2]. Other methods [19] have been developed using heuristics that takes into account a set of potential text features like term frequency, location of words, cue method, sentences length, and proper nouns to identify relevant information in text. Unsupervised learning methods use graph based models [20] that consider sentences salience based on eigenvector centrality in a graph representation of sentences; concept oriented methods that uses concepts extracted from an external knowledge base; and fuzzy logic that evaluates a set of document features like presence of proper nouns, sentences length, sentences position, sentence to sentence similarity [21].

Recent methods use supervised learning to train models to generate summaries given a large training corpus of labeled summaries. These methods generally use Recurrent Neural Networks, such as Long-Short Term Memory (LSTM) with attention mechanisms to train the summarization function [15]. More complex neural network architectures implement bidirectional Recurrent Neural Networks that operate at the sentence and word levels simultaneously [16], improving the learning process performance. Recent methods add reinforcement learning mechanisms based on ROUGE scores to improve classification [17].

Nallapati *et al.* [16] develops a recurrent network based sequence classifier extractive summarization called SummaRuNNer. In their approach, each sentence is visited sequentially in the original document order and classified as either belonging to the summary or not. Their model contains a two-layer bidirectional Gated Recurrent Unit (GRU-RNN), where the first layer takes into account the word information and the second layer takes into account the representation of the sentences in the document. Using CNN/Daily Mail dataset, they achieve a F-score of 0.39 for ROUGE-1, 0.16 for ROUGE-2 and 0.35 for ROUGE-L measures.

Narayan *et al.* [17] argue that using only the cross-entropy training is not optimal for extractive summarization, since it tends to generate verbose summaries with unnecessary information. Hence, they propose to globally optimize the ROUGE evaluation metric and learn to rank sentences through a reinforcement learning objective. The neural summarization model combines the maximum likelihood cross-entropy loss with rewards from policy gradient reinforcement learning in order to optimize the evaluation metric relevant for the summarization task, *i.e.*, the ROUGE score. The model called REinFoRcement Learning-based Extractive Summarization (REFRESH) was tested in the CNN/Daily Mail dataset, achieving F-score of 0.40 for ROUGE-1, 0.18 for ROUGE-2 0.36 for ROUGE-L.

Zhou *et al.* [22] propose a joint sentence scoring and

selection model for extractive document summarization called NEUSUM. Their model is composed by a bi-directional Gated Recurrent Unit (BiGRU) to encode sentences as a sequence of words, and a BiGRU to encode documents as a sequence of sentences. A final GRU is trained to remember the partial output summary, taking into account the previous sentence in the summary. Using CNN/Daily Mail dataset, NEUSUM achieves F-score of 0.41 for ROUGE-1, 0.19 for ROUGE-2, and 0.37 for ROUGE-L.

Zhang *et al.* [18] propose an approach to summarize texts using latent representations as sentences. The authors describe a three-level architecture, where the first level is a sentence encoder, which converts a sentence into a latent representation by processing the sentence words in a Bidirectional LSTM. The second level, the document encoder, converts the sentences into a document representation by applying a new Bidirectional LSTM. Finally, the document encoder receives the sentence representations and classifies each of them as belonging to the summary or not. Using the selected sentences, they train a model to approximate the selected sentences to the ideal ones in the groud-truth. Using the CNN/Daily Mail dataset, they obtain an F-score of 0.41 for ROUGE-1, 0.18 for ROUGE-2, and 0.37 for ROUGE-L.

VII. Conclusion

Our Vector Offset Summarization (VOS) is based on word embeddings ability to capture the syntactic and semantic relationship between words. Our method succeeds in identifying key aspects of texts, is easy to implement and interpret, and requires much less data than the related approaches described in Section VI. A major drawback of this method is that it is dependent of an abstractive ground truth value to generate summary, working in a similar manner as question-answering solutions, which makes VOS difficult to use in practical summarization tasks.

Using different word embedding models with different dimension, we observed that results are really close but not proportional to the dimensionality of the embeddings. Given that the embedding model with higher dimension has less than half the number of words in vocabulary, we conjecture that a large number of out-of-vocabulary words may bias the final results of VOS-600 in comparison to VOS-100.

Manually evaluating results, we identified that in some texts VOS yields better results than ROUGE scores from a subjective point of view, when we are concerned about the predominant content in a text, without taking into account ground truth values. Ground truth values seem to bias the expected result when using ROUGE scores, ignoring predominant content in text in favor of ground truth input.

Giving its simplicity, VOS accuracy is totally dependent on the quality of the word embeddings. We envision that we can improve performance by refining word embeddings training with high dimensional vectors and larger vocabulary, and training word embeddings over domain specific datasets.

For future work, we want to develop a method that generalizes the relation between a text vector and its ground truth vector, overcoming the ground truth requirement for each text. We also want to address redundancy in summaries using clustering algorithms to identify key distinct ideas in text and select for each cluster the most representative sentence.

Acknowledgements: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) agreement (DOCFIX 04/2018). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017.

[2] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[3] V. Dalal and L. G. Malik, "A survey of extractive and abstractive text summarization techniques," in *ICETET 2013*, 2013, pp. 109–110.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS 2013*, 2013, pp. 3111–3119.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP'14*, 2014, pp. 1532–1543.

[7] J. P. Aires, D. Pinheiro, V. S. de Lima, and F. Meneguzzi, "Norm conflict identification in contracts," *Artificial Intelligence and Law*, vol. 25, no. 4, pp. 397–428, 2017.

[8] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[9] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[10] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," in *NAACL–HLT 2018*, 2018, pp. 528–540.

[11] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *NAACL-HLT 2013*, 2013, pp. 746–751.

[12] O. Levy and Y. Goldberg, "Linguistic regularities in sparse and explicit word representations," in *CoNLL 2014*, 2014, pp. 171–180.

[13] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *NAACL '03*, 2003, pp. 71–78.

[14] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *NIPS'15*, 2015, pp. 1693–1701.

[15] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *ACL 2016*, 2016, pp. 484–494.

[16] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *AAAI'17*, 2017, pp. 3075–3081.

[17] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," in *NAACL—HLT 2018*, 2018, pp. 1747–1759.

[18] X. Zhang, M. Lapata, F. Wei, and M. Zhou, "Neural latent extractive document summarization," in *EMNP 2018*, 2018, pp. 779–784.

[19] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *SIGIR '95*, 1995, pp. 68–73.

[20] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, 2004.

[21] N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," in *ICCCSP 2017*, 2017, pp. 1–6.

[22] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *ACL 2018*, 2018, pp. 654–663.