Multi-Document Summarization of Persian Text Using Paragraph Vectors

Morteza Rohanian
University of Tehran
rohanian@ut.ac.ir

Abstract

A multi-document summarizer finds the key topics from multiple textual sources and organizes information around them. In this paper we propose a summarization method for Persian text using paragraph vectors that can represent textual units of arbitrary lengths. We use these vectors to calculate the semantic relatedness between documents, cluster them to a number of predetermined groups, weight them based on their distance to the centroids and the intra-cluster homogeneity and take out the key paragraphs. We compare the final summaries with the gold-standard summaries of 21 digital topics using the ROUGE evaluation metric. Experimental results show the advantages of using paragraph vectors over earlier attempts at developing similar methods for a low-resource language like Persian.

1 Introduction

Multi-document summarization is the task of taking the most important points of multiple input documents and put them forward in a short, cohesive way that is easy to follow. The main goal of multi-document summarization is to provide an outcome as objective as possible that can represent all documents with emphasis on key topics. Today, automatic summarization is being used as a technology that makes huge amount of online data accessible in a time that is not practicable to produce summaries manually.

Automatic summarization has been an active field of study in recent years and there are variety of approaches that have been proposed so far. Roughly, summarization methods can be split into two classes: abstractive summarization in which the system analyses the input documents and generates summaries; and extractive summarization which is the method of selecting parts of the input documents and employing them without any changes.

Extractive summarization approaches can be based on supervised learning, using document-summary pairs as training data in order to predict textual units worthy of being in the summary. These textual units can be keywords (Hong and Nenkova, 2014), phrases and sentences (Hu and Wan, 2013). Employing cluster centroids was one of the first attempt to use an unsupervised learning method for summarization (Radev et al., 2004). Sentences are represented as weighted vectors and put in clusters by applying cosine similarity. Pre-trained word vectors also have been used for extractive summarization (Kageback et al, 2014). For sentence representation the authors apply 2 different approaches of adding word vectors and unfolding recursive auto-encoder (RAE).

Our method employs paragraph vector models to compute semantic relatedness between fixed-sized textual units with variable lengths (different paragraphs) and uses K-means clustering algorithm for grouping them based on their proximity. We use the centroids of clusters as the main topics of documents and choose representative paragraphs according to their weights.

2 Related Work

Based on statistical information derived from word frequency, Luhn (1958) was first to propose an algorithm for automatic extractive summarization. Later works expanded Luhns word based approach by adding other weighted statistical or linguistic features of documents. Kupiec et al. (1995) present a supervised method that considered sentences with the title-keywords in them as
the most important ones.

Radev et al. (2004) use centroids of news articles clusters as the most relevant topics for extractive summarization. Bonzanini et al. (2013) use variety of distance metrics for finding representative sentences and develop an algorithm for removing unimportant sentences.

After the idea of distributed word representations was proposed, it has been applied to many natural language processing tasks and demonstrated a great deal of advantage over the traditional textual representations. Kageback et al. (2014) use continuous vector representation as a basis for measuring similarity of sentences. The authors evaluate different settings of word and phrase embeddings and similarity measures on a dataset of 51 topics with corresponding abstractive summaries. Zhang et al. (2015) having word vectors as features, employ a context window to take word order into account. Their method is developed based on the Extreme Learning Machine (ELM).

There have been a few studies for Persian text summarization. Shakeri et al. (2012) use a graph theory approach to find the most important sentences of any given document. Honarpisheh et al. (2008) present a method based on a dictionary and a word segmentation system that transforms collections of documents into a matrix and takes out the most important sentences from the most important clusters.

3 Method

In this section we present our multi-document summarization method which mainly consists of two components: our two different approaches for paragraph representation and our clustering process.

3.1 Vector Representation of Paragraphs

Word embedding is a language modeling technique in which words are represented by vectors of real numbers (Mikolov et al., 2013a). The vectors are obtained by training the neural network on the basis of a text corpus and can have different number of parameters. The parameters are used to predict neighboring words of a given word by getting updated in different context. It has been shown that the final vectors represent the semantic and syntactic relationships between words much better than any other language modeling techniques. According to Bengio et al. (2003) the word vectors are generally based on three ideas:

1- Each word in the body of an n-dimensional feature vector containing the corresponding real numbers.

2- Joint probability function for words getting represented using these vectors.

3- Feature vectors learning and probability function parameters learning happening at the same time.

For providing an extractive summary we need to somehow represent sentence and paragraph vectors. A simple way is to just add or average the word vectors to make textual units vectors bigger than words. Here, we learn vector representations from a given corpus and generate the new fixed-length, n-dimensional vectors by adding all n-dimensional word vectors of a paragraph (Mikolov et al., 2013a). When paragraphs with varying length are having vectors with same dimensions, it is possible to measure their similarities. The main problem with this method is that we ignore the word order just like traditional bag-of-words models. In recent years there have been models that in addition to words and sentences, can represent paragraphs and large documents and do not need task-specific tuning of word weightings functions. Paragraph Vector is an unsupervised framework that concatenates the textual vectors with variable lengths with several word vectors and estimate the next word in a given context. They can be basis of similarity measurement between different sentences, paragraphs and documents (Le and Mikolov, 2014).

In Paragraph Vector method, each unique paragraph is represented by a vector as a column in a matrix. Each word is represented by a vector as a column in another matrix. As shown in Figure 1, paragraph and word vectors are used to predict other words within the documents by either concatenating or averaging. The generated vectors can be applied as the features of different classification or clustering techniques like K-means.

3.2 Clustering Paragraph Vectors

For extracting the most important paragraphs, first we need to group the most similar paragraphs into K clusters and capture both the central and far out paragraphs from each cluster. Each paragraph belongs to the cluster with the nearest mean, and the most central paragraph of each cluster is the
main paragraph of that cluster. Here, we cluster close paragraphs by computing semantic similarities using specific number of latent topics within them. The most distant paragraph from the center in each cluster is the outer point of the cluster. More weight around the center of cluster shows that the central paragraph is a good representation of that cluster and we don't need to add outer paragraphs for a more balanced point of view.

To find the main paragraphs, first we initialize the K centroids randomly. Each paragraph vectors is assigned to its nearest centroid, using a specific distant metric. Each cluster center vector acts as the representation of that cluster. Then we compute the new centroids by taking the mean of all vectors assigned to each cluster. The algorithm stops until no paragraph vectors change clusters.

We use cosine similarity, which particularly used for non-zero vectors and Euclidean metric for an n-dimensional space. We weight paragraphs based on the density of each cluster. As proposed by Leskovec et al., (2014) if the ratio of the number of paragraphs divided by some power of the diameter of a cluster is below some threshold we can consider that cluster as a separated, and extract the outer point in addition to central paragraph. We use the number of dimensions of the space as the power. The number of all paragraphs in our documents divided by dimensions number power of the diameter of our vector space is our threshold. The diameter is still the maximum distance between any two points in the cluster.

3.3 Evaluation Method

For evaluation we use ROUGE (Lin, 2004), which is a recall-oriented, BLEU-like tool that considers n-gram overlaps between the automatic summaries and set of manually-provided, gold-standard summaries. There are three different ROUGE scores that are the common measurement tools used in recent publications: ROUGE-1, ROUGE-2, and ROUGE-SU4, which count matches in unigrams, bigrams, and skip-bigrams. ROUGE-1 and ROUGE-2 have the strongest association with the gold-standard summaries.

We used standard recall, precision and F-measure for reporting the relevance of summaries. Recall is the percentage of captured manually provided summaries and precision is the percentage of summaries that are in the gold-standard data. The F-measure is a weighted average of precision and recall and the outcome is bounded between 0 and 1.
4 Experiments

4.1 Dataset

In order to evaluate our method we provide a dataset of 21 digital topics with each topic having between 50 and 150 paragraphs made by various users which were gathered from different Persian review sites. Each topic is about a specific property of a digital product (e.g. Screen of iPhone SE or Battery of Nikon D7200). The dataset includes gold-standard abstractive summaries of 3 for each topic.

4.2 Parameter Settings

The model in our experiment is trained on Persian Wikipedia dataset of 2.5 GB using fastText (Bojanowski et al., 2016). The dimension of vectors was considered 100 and 200 for words using Skip-gram neural networks model and 50 and 100 for paragraphs. Our models were trained 50 and 100 epochs. We chose 3 and 5 as the number of clusters and used two different distance metrics. We conduct our experiments on a 4.0GHz computer.

4.3 Results and Discussion

The multi-document summarization methods described in Section 3 evaluation results with different parameters are shown with varying parameters in Table 1. The results based on word vectors from the ROUGE-1, ROUGE-2, and ROUGE-SU4 scorings show that our model performs better with higher dimensions and smaller K. The model earns the maximum F-measure of 24.1 on word vectors and gains from training for a greater number of epochs. The results are stable among various ROUGE scores which means that the outputs with most correlation with human-provided summaries also are more fluent.

As can be seen from the Table, paragraph vectors reached the best performances for both three and five clusters compared to word vectors. Using cosine similarity measure improves the recall for different ROUGE scores evaluations while models using Euclidean metric have higher precision and F-measure. High recalls and F-measures are not achievable for our extractive summarization method due to our gold-standard summaries being abstractive.

Indicated by the results Paragraph Vector appears to be a better representation to use for clustering semantically related documents, likely because of word order consideration. The ROUGE-SU4 scores also show that in addition to semantics, paragraph vectors are more effective in capturing syntactic information.

Table 1 shows that capturing the outer topics by weighting paragraphs leads to a more balanced summary and a better ROUGE-SU4 score. It is more convincing for an individual user to read a summary from multiple perspectives. We do not observe significant improvements in ROUGE-1 and ROUGE-2 scores of our weighted models.

5 Conclusion

We described a method clustering main topics of several documents using paragraph vectors as the basis of semantic similarity between paragraphs of different lengths. The centroids of clusters considered as the main topic and paragraphs were weighted based on the distribution of data in each group. The paragraphs with more weight were extracted and compared to human-provided abstractive summaries.

The results suggest that weighting the paragraphs can make summaries more objective. However, the weighting can be combined by several other textual semantic features like the polarity of paragraphs. Considering coreference resolution also makes the summaries more coherent and decrease the level of redundancy.

References

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. Journal of Machine Learning Research,
### ROUGE-1

| Model                  | Dimensions | Epochs | K       | Similarity | Precision | Recall | F-measure |
|------------------------|------------|--------|---------|------------|-----------|--------|-----------|
| Word Vector            | 100        | 50     | 3       | Cosine     | 21.3      | 26.3   | 23.5      |
| Word Vectors_w         | 100        | 50     | 3       | Euclidean  | 22        | 26.6   | 24.1      |
| Word Vector            | 100        | 50     | 5       | Euclidean  | 21.5      | 25.9   | 23.5      |
| Word Vectors_w         | 100        | 50     | 5       | Cosine     | 19.7      | 23.3   | 21.3      |
| Word Vector            | 200        | 100    | 3       | Cosine     | 20.5      | 24.3   | 22.2      |
| Word Vectors_w         | 200        | 100    | 3       | Euclidean  | 21.6      | 26.1   | 23.6      |
| Word Vector            | 200        | 100    | 5       | Euclidean  | 20.5      | 25.6   | 22.8      |
| Paragraph Vectors      | 50         | 50     | 3       | Cosine     | 23.5      | 28.9   | 25.9      |
| Paragraph Vectors_w    | 50         | 50     | 3       | Cosine     | 23.4      | 31.5   | 26.9      |
| Paragraph Vectors      | 100        | 100    | 5       | Euclidean  | 23.1      | 28.7   | 25.8      |
| Paragraph Vectors_w    | 100        | 100    | 5       | Euclidean  | 23.5      | 28.7   | 25.8      |

### ROUGE-2

| Model                  | Dimensions | Epochs | K       | Similarity | Precision | Recall | F-measure |
|------------------------|------------|--------|---------|------------|-----------|--------|-----------|
| Word Vector            | 100        | 50     | 3       | Cosine     | 2.8       | 5.3    | 3.7       |
| Word Vectors_w         | 100        | 50     | 3       | Euclidean  | 3.8       | 5.6    | 4.5       |
| Word Vector            | 100        | 50     | 5       | Euclidean  | 3.2       | 5.2    | 4         |
| Word Vectors_w         | 100        | 50     | 5       | Cosine     | 3.3       | 6.1    | 4.3       |
| Word Vector            | 200        | 100    | 3       | Cosine     | 3.5       | 5.8    | 4.4       |
| Word Vectors_w         | 200        | 100    | 3       | Euclidean  | 3.2       | 5.9    | 4.1       |
| Word Vector            | 200        | 100    | 5       | Euclidean  | 3.3       | 5      | 4         |
| Paragraph Vectors      | 50         | 50     | 3       | Cosine     | 4         | 7.3    | 5.2       |
| Paragraph Vectors_w    | 50         | 50     | 3       | Cosine     | 4.3       | 8.2    | 5.6       |
| Paragraph Vectors      | 100        | 100    | 5       | Euclidean  | 4.4       | 7.3    | 5.5       |
| Paragraph Vectors_w    | 100        | 100    | 5       | Euclidean  | 4.1       | 7.9    | 5.4       |
| Paragraph Vectors_w    | 100        | 50     | 3       | Cosine     | 4.3       | 8.4    | 5.7       |

### ROUGE-SU4

| Model                  | Dimensions | Epochs | K       | Similarity | Precision | Recall | F-measure |
|------------------------|------------|--------|---------|------------|-----------|--------|-----------|
| Word Vector            | 100        | 50     | 3       | Cosine     | 6.2       | 9.1    | 7.4       |
| Word Vectors_w         | 100        | 50     | 3       | Euclidean  | 6.3       | 9.9    | 7.7       |
| Word Vector            | 100        | 50     | 5       | Euclidean  | 5.5       | 8.9    | 6.8       |
| Word Vectors_w         | 100        | 50     | 5       | Cosine     | 6.7       | 10.2   | 8.1       |
| Word Vector            | 200        | 100    | 3       | Cosine     | 6.7       | 8.4    | 7.5       |
| Word Vectors_w         | 200        | 100    | 3       | Euclidean  | 6.2       | 9.7    | 7.6       |
| Word Vector            | 200        | 100    | 5       | Euclidean  | 6.5       | 8.4    | 7.3       |
| Paragraph Vectors      | 50         | 50     | 3       | Cosine     | 7.9       | 10.1   | 8.9       |
| Paragraph Vectors_w    | 50         | 50     | 3       | Cosine     | 8         | 12.4   | 9.7       |
| Paragraph Vectors      | 100        | 100    | 5       | Euclidean  | 7.6       | 10.4   | 8.8       |
| Paragraph Vectors_w    | 100        | 100    | 5       | Euclidean  | 8         | 12.3   | 9.7       |
| Paragraph Vectors_w    | 100        | 50     | 3       | Cosine     | 8.3       | 12.3   | 9.9       |

Table 1: ROUGE scores of extractive summaries using paragraph and word vectors.
Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

Marco Bonzanini, Miguel Martinez-Alvarez, and Thomas Roelleke. 2013. Extractive summarization via sentence removal: Condensing relevant sentences into a short summary. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 13, pages 893-896. ACM.

Mohamad Ali Honarpisheh, Gholamreza Ghassem-Sani, and Seyed Abolghasem Mirroshandel. 2008. A Multi-Document Multi-Lingual Automatic Summarization System. In IJCNLP, pages 733-738.

Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In Proceedings of EACL.

Yue Hu and Xiaojun Wan. 2015. Ppsgen: learning-based presentation slides generation for academic papers. IEEE transactions on knowledge and data engineering 27, no. 4: 1085-1097.

Mikael K Lageback, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive summarization using continuous vector space models. In CVSC at EACL, pages 3139.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pages 68-73. ACM.

Quoc Le, and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1188-1196.

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

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. IBM Journal of research and development 2, no. 2 : 159-165.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. ArXiv preprint arXiv:1301.3781

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111-3119.

Dragomir R Radev, Hongyan Jing, Magorzata Sty s, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. Information Processing and Management, 40(6):919938

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. Mining of massive datasets. Cambridge University Press.

Hassan Shakeri, Saeedeh Gholamreza-Azedeh, Mohsen Amini Salehi, and Fatemeh Ghadamyari. 2012. A New Graph-Based Algorithm for Persian Text Summarization. In Computer Science and Convergence, pages. 21-30.

Yong Zhang, Meng Joo Er, and Rui Zhao. 2015. Multi-Document Extractive Summarization Using Window-Based Sentence Representation. Computational Intelligence, 2015 IEEE Symposium Series on. IEEE.