EASY: Evaluation System for Summarization

Marina Litvak, Natalia Vanetik, and Yael Veksler

Department of Software Engineering
Shamoon Engineering College
Beer Sheva, Israel
{marinal, natalyav, yaelva}@sce.ac.il

Abstract. Automatic *text summarization* aims at producing a shorter version of a document (or a document set). *Extractive* summarizers compile summaries by extracting a subset of sentences from a given text, while *abstractive* summarizers generate new sentences. Both types of summarizers strive to preserve the meaning of the original document as much as possible. Evaluation of summarization quality is a challenging task. Due the expense of human evaluations, many researchers prefer to evaluate their systems automatically, with help of software tools. Automatic evaluations are usually performed to provide comparisons between a system-generated summary and one or more human-written summaries, according to selected measures. However, a single metric cannot reflect all quality-related aspects of a summary. For instance, evaluation of an extractive summarizer by comparing, at word level, its summaries to the abstracts written by humans is not good enough. This is so because the summaries being compared do not necessarily use the same vocabulary. Also, considering only single words does not reflect the coherency or readability of a generated summary. Multiple tools and metrics have been proposed in literature for evaluating the quality of summarizers. However, studies show that correlations between these metrics do not always hold. In this paper we present the **E**v**A**luation **SY**stem for Summarization (EASY), which enables the evaluation of summaries with several quality measures. The EASY system can also compare system-generated summaries to the extractive summaries produced by the OCCAMS baseline, which is considered the best possible extractive summarizer. EASY currently supports two languages–English and French–and is freely available online for the NLP community.

1 Introduction

Automatic *text summarization* aims at representing a text document (or a document set) in a short concise form. The output of text summarization for a given text is called a *summary*. The size of a summary is usually limited by a predefined number of words or sentences. A summary can be either generic or tailored to fit the user's needs. A generic summary is expected to convey the meaning of the whole text while a tailored summary is expected to reflect the interests of a user; statements of the user's interests can come in many forms, including

those of query, subject, and style. Text summarization is important in the age of increasing volume of information that is available on-line. Several extensive surveys of automatic summarization can be found at [1–3].

Automatic text summarization approaches can be divided into two main categories. Extractive summarization [4],[5] deals with selecting a subset of sentences from the original document(s) without modifying them. Abstractive summarization can compile summaries by extracting parts of original sentences (also called compressive summarization [5]) or by generating new sentences word by word [6], using a different vocabulary.

The need for quality assessment of summarization tools is obvious. Using human evaluators is extremely time-consuming and labor-intensive. Additional issues arise when using this approach, such as the qualification of evaluators and their agreement on a content of generated summaries. [7] Also, hiring qualified evaluators to work with multilingual domain is not an easy task.

Therefore, there is an existing need to acquire automatic tools for summary evaluation. Moreover, such tools must provide a wide range of metrics for covering multiple quality aspects, such as the informativeness (or relevance) of a summary, coverage of the main topics of a document, and the coherency and readability of the summary.

Automatic evaluation relies on comparison between the summaries generated by an automatic system (*system summaries*) and summaries that have been produced by humans; these are called *gold standard summaries* or *reference summaries*. These summaries may be created from scratch by humans or produced by merging several human-produced summaries by using the majority rule[8]. In both cases, reference summaries usually contain new sentences that are not present in an original document. When reference summaries are not available, system summaries may be compared to original texts through the use of a metric that helps to see how information in the whole text is covered by a summary[9]. Results of automatic evaluation depend closely on the chosen metric.

Papers [10] and [9] contain surveys of early evaluation measures for text summarization. Paper [11] gives an overview of different methods for evaluating automatic summarization systems, and describes different evaluation criteria such as coherence, informativeness, different scoring approaches, and means of analyzing summary content.

Following [10] and [12], summarization evaluation methods can be divided into two categories: extrinsic evaluation, where the summary quality is judged by its helpfulness for a given task, and intrinsic evaluation, where a summary is analyzed directly. Our study focuses on intrinsic evaluation of generic summaries (where no user queries are supplied).

We can roughly assign all intrinsic evaluation methods to the methods comparing between system and human summaries and the methods comparing between system summaries and their documents. The metrics provided in the first category measure the closeness (similarity) of the generated summary to reference summaries that represent the ideal summaries, while the metrics calculated in the second category measure the summary's coverage of the main topics de-

scribed in a document. We will call the first category "similarity" and the second one "coverage." While the "similarity" methods can be performed in either the lexical (i.e., words) or semantic (i.e., topics) level, comparison between a summary and its document in the lexical level is meaningless. Therefore, for measuring coverage of topics in a generated summary, semantic text representation must be utilized.

There are multiple metrics that compare between system and reference summaries in the lexical level. These metrics measure the similarity between vocabularies [13] of summaries. Some of them are applicable to extractive summarization only, such as metrics based on sentence recall or precision [14–16], or metrics that rely on sentence rank (in terms of summary-worthiness); they measure the correlation between sentence sequences representing system and reference summaries [17].

The Bleu machine translation evaluation measure [18] has been used as a summarization metric in [19].

Metrics in the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) family, proposed in [20], count the number of overlapping units such as n-grams, word sequences, and word pairs between the system and the reference summaries. This remains the most popular metric for summarization evaluation. In [21], the authors present the Merge Model Graph (MeMoG) metric for evaluating summaries, which uses n-gram graphs for comparing system and reference summaries. Tests on summaries produced for MultiLing-2015 tasks [22] have shown a clear indication that the MeMoG is much less sensitive than ROUGE to differences in text preprocessing. Both tools are also applicable to evaluation of abstractive summaries, but, as all lexical-based methods, they do not consider semantic similarity between system and reference summaries.

An alternative solution to the lexical comparison between system and reference summaries is to consider their semantics. The Pyramid method discussed in [23] involves semantic matching of content units, to which differential weights are assigned based on their frequency in a corpus of summaries. Semantic models such as latent semantic analysis (LSA)[24], topic modeling with latent Dirichlet analysis (LDA)[25], word embeddings with Word2Vec[26] and Doc2Vec[27], are also popular for comparing summaries to reference summaries or to original documents. This is because they represent texts as semantic vectors. In [12] the authors propose an LSA-based evaluation measure and show its high correlation to human rankings. In [28] and [29] word embeddings were shown as a good means for evaluating summaries.

Attempts to create a platform for summary evaluation have been previously made. The SUMMAC system [30] provided the first system-independent framework for summary evaluation. It included several extrinsic and intrinsic methods for evaluating summaries. In the extrinsic categorization task, an evaluation is made by finding whether there is enough information contained in a summary to provide successful categorization of the document. In an intrinsic question-answering task a topic-related summary for a document was evaluated in terms

of its 'informativeness,' namely, the degree to which it contained answers to a set of topic-related questions.

Paper [31] described a framework in which various automated summary content evaluation methods can be situated, and implemented a specific variant that uses short text fragments (called Basic Elements). Multiple similarity metrics were introduced and their correlations with other known metrics, such as ROUGE, were reported. Most introduced metrics are lexical-based, except one that applied synonym resolution using WordNet. In [32] the authors present a summarization assessment system that does not rely on reference summaries. There, a coverage metric was proposed as a combination of syntactic (words order) and semantic (using WordNet) information of sentence words.

In this paper we introduce a very preliminary version of the **Eva**luation **Sy**stem for Summarization; a system we have named EASY. We have designed EASY for evaluation of summarization results and ranking summarization tools. At its current state, the system enables the user to evaluate the quality of generic (i.e., unrelated to specific query or topic) summaries using two different types of metrics–ROUGE [20] and MeMoG [21]. As such, EASY currently supports only similarity metrics at the lexical level. It also enables users to compare the scores of evaluated summaries to corresponding scores of summaries that were produced by two baseline methods: TopK, which selects first sentences and the Optimal Combinatorial Covering Algorithm for Multi-document Summarization (OCCAMS) [33], which produces 'ideal' extractive summaries. EASY also enables the user to view the correlation between scores that have been produced by the application of different metrics. EASY is freely available online and open for use by anyone in NLP community.

This paper is organized as follows. Section 2 describes the summarization metrics used by and the baseline summarizers implemented in EASY. Section 3 shows and explains EASY's interface. Sections 4 and 5 address the system's availability and give directions for its further development.

2 EASY system design

In this section we describe the capabilities of the EASY system and the algorithms it implements. To evaluate the summaries, the system needs the following input from the user.

1. A **folder containing original documents** in UTF-8 text format, where every document is stored in a separate file. In case of multi-document summarization, every document set should be merged into a single file.
2. A **folder containing gold standard summaries** (called **reference summaries**) should be available, with one or more for every summarized document. A document and its reference summaries are matched by a case-sensitive their name parts before the file extension.
   - For example, a document named "**test**.[ext]" will be matched to two reference summaries named "**test**.A.[ext]" and "**test**.B.[ext]", where [ext] is any extension.

– Different reference summaries are distinguished by their first extension, i.e. "**test**.A1.[ext]" and "**test**.A2.[ext]" are treated as two different reference summaries for the same document "**test**.[ext]".

3. A **folder containing summaries being evaluated**, with one summary for each document. A document and its summary are matched by a case-sensitive comparison of their name parts before file extension. For example, document named "**test**.[ext]" will be matched to summary "**test**.[ext]".

When input documents and summaries are supplied, the user can select metrics (described in Section 2.1) and their parameters, and can also choose whether to make comparison to baselines (described in Section 2.2), and then engage in the evaluation process. During this process, baseline summaries are computed and then both baseline summaries and system summaries are compared to reference summaries (gold standard) through the use of metrics and parameters chosen by the user. The general pipeline of the EASY system is depicted in Figure 1. A detailed user story is described in Section 3.

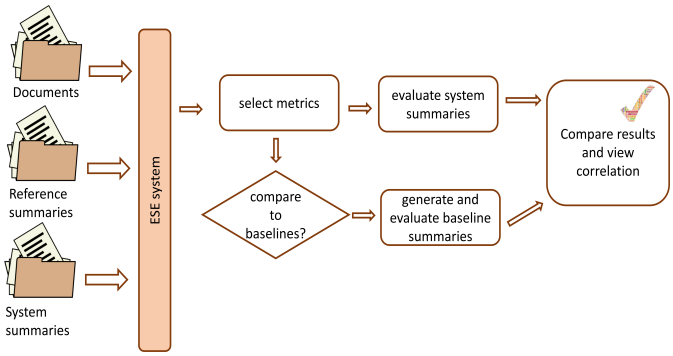

Fig. 1. EASY system flow.

2.1 Summarization quality metrics

In this section, we explain how summarization metrics are used in our system.

2.1.1 ROUGE metrics

Paper [20] presented set of metrics called ROUGE that is used for evaluating automatic summarization in NLP. ROUGE represents a set of similar metrics such as ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU. Its main idea is to count overlapping units (such as n-grams, word sequences and word pairs) between a system summary and reference summaries. Intuitively, higher ROUGE scores show that the system summary is of higher quality. This

metric is currently the most popular metric of its type that is in use, especially in the field of text summarization (see [34]). In our system, we implemented several ROUGE metrics, described below.

1. ROUGE-N, which measures overlap of n-grams between the system summary S and reference summaries $R = \{r_1, \ldots, r_k\}$ with a user-defined n that is usually set to a number between 1 and 4.

   (a) recall-based ROUGE-N is computed as

   $$R_{n\text{-}grams} = \frac{\sum_{1 \le i \le k} \sum_{n\text{-}gram \in r_i} count_{match}(n\text{-}gram)}{\sum_{1 \le i \le k} \sum_{n\text{-}gram \in r_i} count(n\text{-}gram)}$$

   where $count()$ is the total number of n-grams, and $count_{match}()$ is the number of common (matching) n-grams.

   (b) precision-based ROUGE-N is computed as

   $$R_{n\text{-}grams} = \frac{\sum_{1 \le i \le k} \sum_{n\text{-}gram \in r_i} count_{match}(n\text{-}gram)}{\sum_{n\text{-}gram \in S} count(n\text{-}gram)}$$

2. Common-subsequence-based metrics include the following

   (a) ROUGE-L, which measures the length of the longest common subsequence $LCS()$ between the system and reference summaries; this measure is an F-measure based as follows:

   $$F_{LCS} = \frac{(1+\beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}}$$

   where β is the system parameter with default $\beta = 1$ (to obtain a harmonic mean of LCS-related recall and precision), and

   $$P_{LCS} = \frac{\sum_{i=1}^{k} LCS(r_i, S)}{|S|}$$

   $$R_{LCS} = \frac{\sum_{i=1}^{k} LCS_{\cup}(r_i, S)}{\sum_{i=1}^{k} |r_i|}$$

   Here, $LCS_{\cup}(r_i, S)$ is the LCS score between system summary S and reference summary r_i that is computed as

   $$LCS_{\cup}(r_i, S) = \cup_{\text{sentence } S_j \in S} LCS(r_i, S_j)$$

   (b) ROUGE-W ([20]), which measures the length of the longest weighted common subsequence and differentiates subsequences by their length. It is computed as an F-measure

   $$F_{WLCS} = \frac{(1+\beta^2) R_{WLCS} P_{WLCS}}{R_{WLCS} + \beta^2 P_{WLCS}}$$

where

$$R_{WLCS} = f^{-1}(\frac{WLCS(S,R}{f(|S|)}),\ P_{WLCS} = f^{-1}(\frac{WLCS(S,R}{f(|r_1| + \ldots + |r_k|)})$$

Function $f()$ is smooth with a smooth inverse, and is usually set to $f(k) = k^2$ so that $f^(-1)(k) = \sqrt{k}$. Parameter β is set to 1 ([35]).

3. Skip-based metrics
   (a) ROUGE-S measures the overlap of skip-bigrams between a candidate summary and a set of reference summaries. It is similar to ROUGE-2 except that a skip-bigram refers to any pair of words in sentence order that allows for arbitrary gaps. The precision and recall are computed as a ratio of the total number of possible bigrams.
   Let $SKIP2(S, r_i)$ denote the number of skip-matches between system summary S and reference summary r_i. Then ROUGE-S is defined as an F-measure based on skip-bigrams

$$R_{SKIP2} = \frac{(1+\beta^2)R_{SKIP2}P_{SKIP2}}{R_{SKIP2} + \beta^2 P_{SKIP2}}$$

   where

$$R_{SKIP2} = \frac{SKIP2(S,R)}{C(|S|,2)},\ P_{SKIP2} = \frac{SKIP2(S,R)}{C(|r_1 + \ldots + r_k|,2)}$$

   and $\binom{C(x,2)=}{x2}$ is the total possible number of bigrams. The maximum skip distance between two words is limited to reduce wrong matches such as: "To to". To ensure this, we define the maximum distance parameter $d_{MAX-SKIP}$ to be 4, so that skip-bigrams are taken into account within the maximum skipping distance only.
   (b) ROUGE-SU measures overlaps of both skip-bigrams and unigrams between a candidate summary and a set of reference summaries. This is because we do not want to assign a 0 score to a candidate summary simply because it does not share a skip bigram with any reference summary when instead it has a common unigram. Therefore, unigrams are added to give credit to the candidate's summary if it does not contain any pair of words with the reference summary.

2.1.2 MeMoG metric

The MeMoG metric, presented in [21], is an evaluation method that based on n-gram graphs. Experimental proof of its high performance for evaluation of summaries in different languages is presented in [22].

Given a set of reference summaries, the MeMoG metric creates an n-gram graph for each of them and an n-gram graph for the system summary. Formally, let $G = \{V, E, W\}$ be an n-gram graph, where V is the set of character n-grams that can be created from the text, E is the set of edges, and W is the weight function that represents the number of times a pair of n-grams is present in a text within a legal distance from each other. This distance is denoted D_{win}. In order to compute this metric, the user should supply the following parameters:

1. L_{min} - minimum length of n-grams,
2. L_{max} - maximum length of n-grams, and
3. D_{win} - the windows size for two n-grams.

The default parameters are $L_{min} = 3$, $L_{max} = 3$ and $D_{win} = 3$, following [21].

Example 1. The n-gram graph for text: "abcb" with $n = 1$, $D_{win} = 1$ is depicted in Figure 2. The text has two co-occurrences of letters 'b' and 'c' and therefore the weight of the edge (b, c) is 2, i.e. $e =< b, c, 2 >$. This edge is undirected, meaning that subtexts 'bc' and 'cb' of the original text are represented by the same edge.

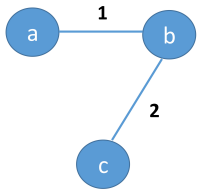

Fig. 2. An n-gram graph.

The next step is to represent all reference summaries by a single n-gram graph [3]. We begin by initializing the graph to be an n-gram graph of any of the reference summaries. The initial graph is then updated using every one of the remaining n-gram reference summary graphs as follows. Let G_1 be the current merged n-gram graph, and let G_2 be the n-gram graph of the next reference summary. The *merge function* $U(G_1, G_2, l)$ defined edge weights as

$$w(e) = w^1(e) + (w^2(e) - w^1(e)) * l$$

where $l \in [0, 1]$ is the learning factor, $w^1(e)$ is the weight of e in G_1, and $w^2(e)$ is the weight of e in G_2. In our system we chose $l = \frac{1}{i}$ where $i > 1$ is the number of the reference graph being processed.

Example 2. Figure 3 shows how edge weight is calculated when merging graph G_1 and reference graph G_2 for the learning factor $l = \frac{1}{3}$, which gives $w(e) = w^1(e) + (w^2(e) - w^1(e)) * \frac{1}{3} = 2.5 + (1 - 2.5) * \frac{1}{3} = 2$

In the MeMoG metric, the score of a summary is one similarity measurement, denoted by VS, between system summary graph G^j and the merged reference graph G^i. The similarity score between edges is defined as

$$VR(e) = {}^{\min\{w^i(e), w^j(e)\}}/_{\max\{w^i(e), w^j(e)\}}$$

where w^i and w^j are weights of the same edge e (identified by its end-node labels) in graphs G^i and G^j respectively. The final score is computed as

$$VS(G^i, G^j) = {}^{\sum_{VR(e)}}/_{\max\{|G^i|, |G^j|\}}$$

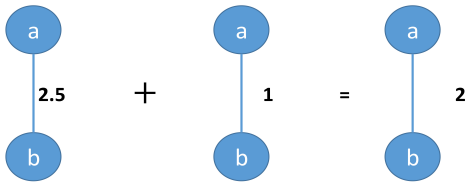

Fig. 3. The merge function of the MeMoG metric.

Example 3. Let two reference summaries be "abca" , "bcab", and let the system summary be "abcab". Figure 4 shows *VS* scores for this system summary with D_{win} set to 1 (1-grams) and 2 (bigrams). Note that setting larger a D_{win} does not necessarily increase the score.

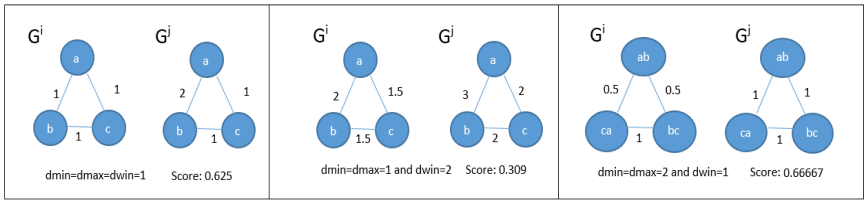

Fig. 4. MeMoG scores for different sizes of n-grams.

2.2 Baselines

2.2.1 TopK baseline

For this baseline, we simply select the first K sentences of the source document so that the number of words of the candidate summary is at least the predefined word limit W, making K minimal. If $K-1$ top sentences contain less than W words, and K sentences contain more than W words, we add the K-th sentence to the summary.

2.2.2 OCCAMS baseline

The OCCAMS, introduced in [33], is an algorithm for selecting sentences from a source document when reference summaries are known. This algorithm finds the best possible sentence subset covering reference summaries because reference summaries are visible to it. While no extractive summary can fully match human-generated abstractive reference summaries, OCCAMS achieves the best possible result (or its good approximation) for the extractive summarization task. Comparing system summaries to the result of OCCAMS shows exactly how far the tested system is from realistic best possible extractive summarization result.

The OCCAMS' parameters are the weights of the terms W, the number of words in sentences C, and the size of the candidate summary L. Let D be the source document consisting of sentences $S_1, \ldots, S_n$ and let $T = \{t_1, \ldots, t_m\}$ be the set of document's terms (tokenized stemmed words). Initially OCCAMS computes document matrix $A = (a_{ij})_{i=1..n, j=1..m}$ using LSA [24] as follows:

$$a_{ij} = L_{ij} \times G_i,\ L_{ij} = \begin{cases} 1, t_j \in S_i \\ 0, t_j \notin S_i \end{cases}$$

G_i is the entropy weight of S_i defined as

$$G_i = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}$$

where p_{ij} is the weight of term t_j in sentence S_i (normalized by total number of appearances of t_j in the document).

Then, OCCAMS computes the singular value decomposition of matrix A as $A = USV^T$, following the approach of [36]. The singular value decomposition produces term weights $w(t_i)$ as follows:

$$w(t_i) = \|(|U_k| \cdot S_k)_i^T\|_1 = \sum_{j=1}^{n} |u_{ij}| \cdot S_{jj}$$

For these weights, the final solution is computed by using Budgeted Maximum Coverage (BMC [37]) and Fully Polynomial Time Approximation Scheme (FPTAS [38]) greedy algorithms to select of sentences that provide maximum coverage of the important terms, while ensuring that their total length does not exceed the intended summary size. The full flow of the OCCAMS algorithm is depicted in Algorithm 1.

Algorithm 1 OCCAMS algorithm

Input: Document D, terms T, sentences C, term weights W
Output: 'Ideal' extractive summary K

1: $K_1 = \text{Greedy\_BMC(T,D,W,C,L)}$
2: $S_{\max} = \text{argmax}_{s_i \in C}\{\sum_{t_j \in S_i} w(t_j)\}$
3: Let $D' = D \setminus S_{\max}$
4: Let $C' = C \setminus S_{\max}$
5: Let $L' = L - |S_{\max}|$
6: Let $T' = T \setminus \{t_i \in S_{\max}\}$
7: $K_2 = S_{\max} \cup \text{Greedy\_BMC}(T', D', W', C', L')$
8: $K_3 = \text{Knapsack}(\text{Greedy\_BMC}(T, D, W, C, 5L), L)$
9: Compute sets of terms $T(K_i)$ covered by solutions K_i, $i = 1, 2, 3$
10: $K = \text{argmax}_{k=1,2,3}\{\sum_{t_j \in T(K_i)} w(t_j\}$

This algorithm closes the gap created following the award of high score by automated assessment system to automated summarization systems.

3 Implementation details

In this section we describe and give examples of the EASY system interface (screen images are taken from standalone implementation[1].

3.1 Input selection

In EASY, a user can make a choice between analyzing a single file with its system and reference summaries, or analyzing an entire corpus. In the former case, the user needs to supply file names for the document, reference summary (or summaries), and the system summary that is to be evaluated. In the latter case, a user needs to select folders that contain a corpus, system summaries and reference summaries. Matching between the document and its corresponding summaries is done by comparing the file name parts that precede file extensions. File names are treated as case-sensitive. Figure 5 shows the input selection interface for the case of a corpus.

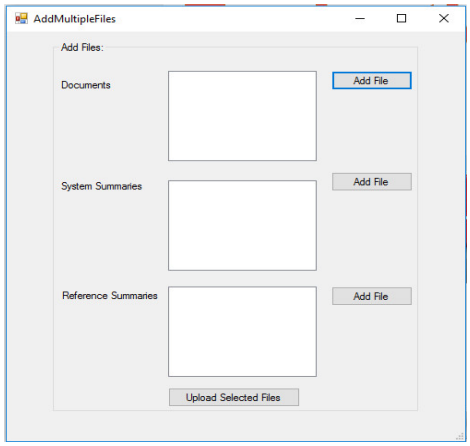

Fig. 5. EASY welcome screen for choosing directories for corpus, system summaries, and reference summaries.

3.2 Metrics

Figures 6 and 7 show how to compute ROUGE and MeMoG summarization metrics for the selected input (corpus, reference summaries, and system summaries). The top part of the interface in both cases enables the user to select parameters for every metric, while the bottom part gives the user an opportunity to compute baseline summaries and to compute the chosen metric for baselines

[1] https://youtu.be/5AhZB5OfxN8

with the same parameters as above. In both cases, the user can choose to work with a single file and not with a corpus, as is shown in the examples.

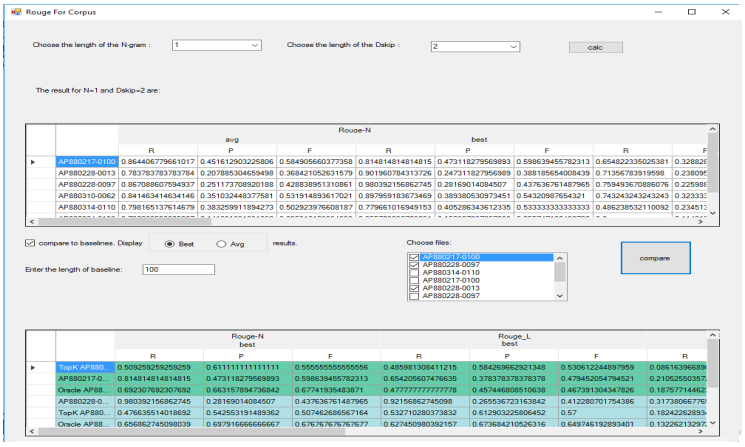

Fig. 6. Computing ROUGE metric for system summaries.

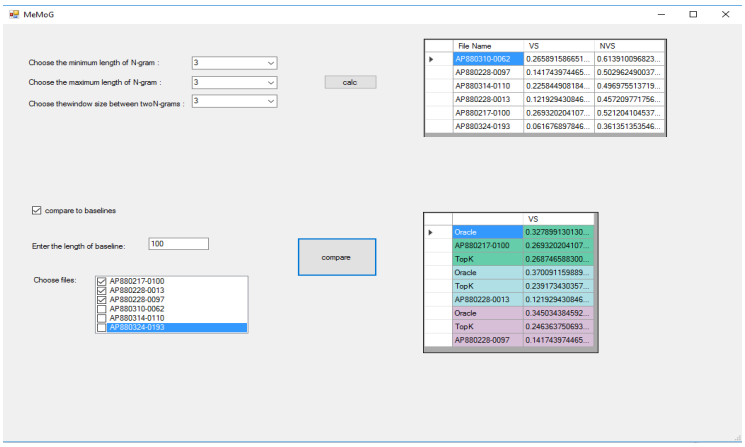

Fig. 7. Computing MeMoG metric for system summaries.

3.3 Baselines

Figures 8 and 9 show how baseline summaries can be generated with the EASY system. The user needs to select one or more files from the loaded corpus and specify the desired summary length (in both examples it is set to 150 words).

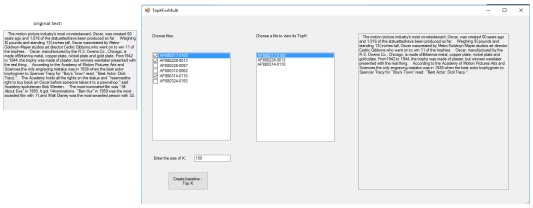

Fig. 8. Summary generation for TopK baseline.

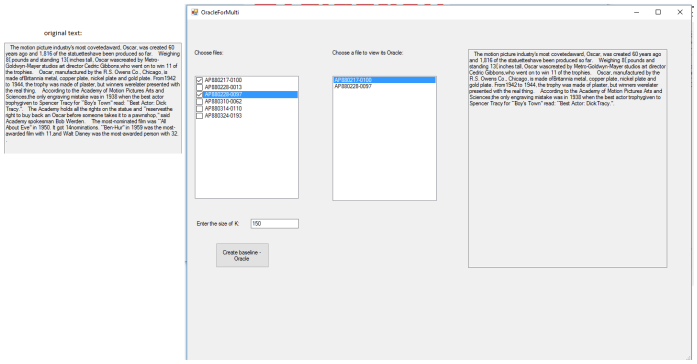

Fig. 9. Summary generation for OCCAMS baseline.

3.4 Correlation of results

The EASY system gives the user the option of computing and viewing Spearman's rank correlation ([39]) between scores obtained for different metrics. The user can select two metrics each time and view visualization of the Spearman correlation as depicted in Figure 10.

4 Availability and reproducibility

The EASY system standalone version is implemented in c#, and its Web version is implemented in Angular7 on the client side, and sp.net WebAPI2 on the server

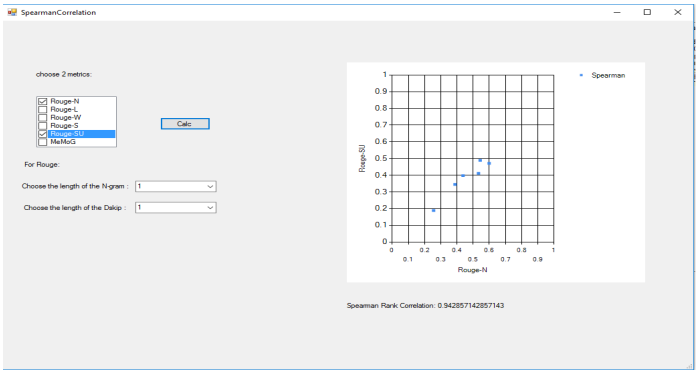

Fig. 10. Spearman's correlation of two metrics' scores.

side. The EASY system is freely available for everyone via its Web interface at https://summaryevaluation.azurewebsites.net/home. Video of the standalone interface operation is available at https://youtu.be/5AhZB5OfxN8. We encourage members of the NLP community to use it for evaluation of extractive summarization results. Currently, the system supports English and French text evaluation only, but in the future we plan to extend it by adding more languages and also by implementing additional metrics.

5 Conclusions

In this paper we present a framework named that we call EASY, which is intended for evaluation of automatic summarization systems. Currently, EASY supports English and French languages. The EASY system enables the users to compute several summarization metrics for the same set of summaries and to observe how they correlate using Spearsman's correlation. The system can also compute baseline summaries using the TopK approach, which takes first sentences of the document, and the OCCAMS approach, which computes optimal extractive summaries by taking into account reference summaries (the gold standard).

In our future work we plan to employ semantic representations based on LSA, topic modeling, and word embeddings, which can be used for implementing both similarity and coverage metrics. Also, we plan to add readability [40] metrics. According to our observations, there are two well-known extractive summarization methods that are widely compared to the new approaches, namely TextRank [41] and integer linear programming optimization [42]. We intend to implement these methods as baseline summarizers in EASY.

Based on our experience, an extensive statistical analysis is usually required for a correct interpretation of results. We intend to provide EASY users with the built-in ability to perform such analysis. We plan to provide an API so that

the members of NLP community will be able to contribute their own implementations of different metrics and baselines.

References

1. Nenkova, A., McKeown, K., et al.: Automatic summarization. Foundations and Trends® in Information Retrieval **5** (2011) 103–233
2. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Mining text data. Springer (2012) 43–76
3. Das, D., Martins, A.F.: A survey on automatic text summarization. Literature Survey for the Language and Statistics II course at CMU **4** (2007) 192–195
4. Gupta, V., Lehal, G.S.: A survey of text summarization extractive techniques. Journal of emerging technologies in web intelligence **2** (2010) 258–268
5. Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artificial Intelligence Review **47** (2017) 1–66
6. Kasture, N., Yargal, N., Singh, N.N., Kulkarni, N., Mathur, V.: A survey on methods of abstractive text summarization. Int. J. Res. Merg. Sci. Technol **1** (2014) 53–57
7. Pittaras, N., Montanelliy, S., Giannakopoulos, G., Ferraray, A., Karkaletsis, V.: Crowdsourcing in single-document summary evaluation: the argo way. In Litvak, M., Vanetik, N., eds.: Multilingual Text Analysis: Challenges, Models, and Approaches. World Scientific (2019)
8. Nanba, H., Okumura, M.: Producing more readable extracts by revising them. In: Proceedings of the 18th conference on Computational linguistics-Volume 2, Association for Computational Linguistics (2000) 1071–1075
9. Jing, H., Barzilay, R., McKeown, K., Elhadad, M.: Summarization evaluation methods: Experiments and analysis. In: AAAI symposium on intelligent summarization, Palo Alto, CA (1998) 51–59
10. Jones, K.S., Galliers, J.R.: Evaluating natural language processing systems: An analysis and review. Volume 1083. Springer Science & Business Media (1995)
11. Mani, I.: Summarization evaluation: An overview (2001)
12. Steinberger, J., Ježek, K.: Evaluation measures for text summarization. Computing and Informatics **28** (2012) 251–275
13. Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, Inc. (1986)
14. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (1995) 68–73
15. Jing, H., McKeown, K.R.: The decomposition of human-written summary sentences. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM (1999) 129–136
16. Merlino, A., Maybury, M.: An empirical study of the optimal presentation of multimedia summaries of broadcast news. Cambridge, MA: MIT Press (1999)
17. Donaway, R.L., Drummey, K.W., Mather, L.A.: A comparison of rankings produced by summarization evaluation measures. In: Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization, Association for Computational Linguistics (2000) 69–78
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on

association for computational linguistics, Association for Computational Linguistics (2002) 311–318
19. Pastra, K., Saggion, H.: Colouring summaries bleu. In: Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: are evaluation methods, metrics and resources reusable?, Association for Computational Linguistics (2003) 35–42
20. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004). (2004) 25–26
21. Giannakopoulos, G., Karkaletsis, V.: Autosummeng and memog in evaluating guided summaries. In: Proceedings of Text Analysis Conference. (2011)
22. Giannakopoulos, G., Kubina, J., Conroy, J., Steinberger, J., Favre, B., Kabadjov, M., Kruschwitz, U., Poesio, M.: Multiling 2015: multilingual summarization of single and multi-documents, on-line fora, and call-center conversations. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue. (2015) 270–274
23. Nenkova, A., Passonneau, R.: Evaluating content selection in summarization: The pyramid method. In: Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004. (2004)
24. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American society for information science **41** (1990) 391–407
25. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
26. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013) 3111–3119
27. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. (2014) 1188–1196
28. Ng, J.P., Abrecht, V.: Better summarization evaluation with word embeddings for rouge. arXiv preprint arXiv:1508.06034 (2015)
29. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International Conference on Machine Learning. (2015) 957–966
30. Mani, I., Klein, G., House, D., Hirschman, L., Firmin, T., Sundheim, B.: Summac: a text summarization evaluation. Natural Language Engineering **8** (2002) 43–68
31. Hovy, E., Lin, C.Y., Zhou, L., Fukumoto, J.: Automated summarization evaluation with basic elements. In: Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006), Citeseer (2006) 604–611
32. Abdi, A., Idris, N.: Automated summarization assessment system: quality assessment without a reference summary. In: The International Conference on Advances in Applied Science and Environmental Engineering-ASEE. (2014)
33. Davis, S.T., Conroy, J.M., Schlesinger, J.D.: Occams–an optimal combinatorial covering algorithm for multi-document summarization. In: Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on, IEEE (2012) 454–463
34. Cohan, A., Goharian, N.: Revisiting summarization evaluation for scientific articles. arXiv preprint arXiv:1604.00400 (2016)
35. Sasaki, Y., et al.: The truth of the f-measure. Teach Tutor mater **1** (2007) 1–5
36. Steinberger, J., Ježek, K.: Text summarization and singular value decomposition. In: International Conference on Advances in Information Systems, Springer (2004) 245–254

37. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. Information processing letters **70** (1999) 39–45
38. Karger, D.R.: A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. SIAM review **43** (2001) 499–522
39. Well, A.D., Myers, J.L.: Research design & statistical analysis. Psychology Press (2003)
40. Lloret, E., Vodolazova, T., Moreda, P., Munoz, R., Palomar, M.: Are better summaries also easier to understand? analyzing text complexity in automatic summarization. In Litvak, M., Vanetik, N., eds.: Multilingual Text Analysis: Challenges, Models, and Approaches. World Scientific (2019)
41. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. (2004)
42. McDonald, R.: A study of global inference algorithms in multi-document summarization. In: Advances in Information Retrieval. (2007) 557–564