CIT. Journal of Computing and Information Technology, Vol. 25, No. 2, June 2017, 149–166
doi: 10.20532/cit.2017.1003398

Mix Multiple Features to Evaluate the Content and the Linguistic Quality of Text Summaries

Samira Ellouze, Mahar Jaoua and Lamia Hadrich Belguith

ANLP research Group, MIRACL Laboratory, University of Sfax, Tunisia

In this article, we propose a method of text summary's content and linguistic quality evaluation that is based on a machine learning approach. This method operates by combining multiple features to build predictive models that evaluate the content and the linguistic quality of new summaries (unseen) constructed from the same source documents as the summaries used in the training and the validation of models. To obtain the best model, many single and ensemble learning classifiers are tested. Using the constructed models, we have achieved a good performance in predicting the content and the linguistic quality scores. In order to evaluate the summarization systems, we calculated the system score as the average of the score of summaries that are built from the same system. Then, we evaluated the correlation of the system score with the manual system score. The obtained correlation indicates that the system score outperforms the baseline scores.

ACM CCS (2012) Classification: Information systems → Information retrieval → Retrieval tasks and goals → Summarization

Computing methodologies → Artificial intelligence → Natural language processing → Information extraction

Computing methodologies → Artificial intelligence → Natural language processing → Discourse, dialogue and pragmatics

Keywords: summary evaluation, content, linguistic quality, machine learning, regression or classification

1. Introduction

With the significant increase in automatic summarization systems, text summary evaluation has become an absolutely necessary task which guides the development of suitable summarization approaches. However, it is a complex task. In fact, the complexity of this task comes from the unclear definition of summarization properties: "What represents a 'good' summary"? It is in this context that several studies have been conducted to develop manual and automatic evaluation metrics of text summaries. These metrics can be divided into intrinsic or extrinsic metrics. Because of the importance of evaluating summarization systems, many evaluation conferences have been organized in the last two decades, such as SUMMAC, DUC (Document Understanding Conference), TAC (Text Analysis Conference), etc., to evaluate the performance of summaries generated automatically. In addition, in the TAC'2009 session, an automatic evaluation task was proposed to encourage researchers to develop automatic evaluation metrics. Most of the metrics previously developed in the field of automatic evaluation of content summaries had focused on the use of surface level analysis (lexical or syntactic). This level does not deal with the use of language phenomena such as synonyms, generalizations, specifications, abbreviations, homographs, etc., in text summaries. For this reason, we need to add other levels of analysis to an evaluation metric. Furthermore, most works have particularly focused on the evaluation of the content and have more or less neglected the linguistic quality evaluation even though [1] has mentioned the importance of this quality to read and understand a text summary easily. In fact, a text summary without reference resolution, with redundant information or with errors in sentence structure cannot be understood. It is in this frame-

work that we have targeted as a field of study both types of evaluation while trying to address some aspects of the semantic level. The initial idea revolves around experiments conducted by [2] and [3], who tried to combine automatic metrics to better correlate with manual metrics. So the objective is to build models able to predict a manual content metric and others able to predict a linguistic quality metric by combining automatic metrics and features defined on the candidate summary. The choice of combining these features as a strategy has a number of advantages. For instance, one can benefit from the use of content features that operate on different levels of analysis. In addition, linguistic quality aggregates several linguistic aspects such as structure and coherence, grammaticality, focus, etc. Those aspects cannot be handled with one simple metric; this is why we have used a combination of features. The combination of features is performed using two machine learning techniques: regression and classification, which will allow us to predict respectively PYRAMID and linguistic quality scores for unseen summary and summarization system. In addition, in the first step, linguistic quality has been evaluated using one predictive model and without taking into account the variation on the topic of each source documents collection. However, the variation of topics leads to a variation in the writing style, the vocabularies used, the structures used, the length of sentences, etc., which may influence the performance of the predictive model. For this reason, in the second step, we evaluated the linguistic quality by building a predictive model for each collection of source documents. The rest of this paper is structured as follows: In Section 2, we present the principal works that have addressed the problem of the evaluation of the content and the linguistic quality of a summary. Then in Section 3, we explain the proposed method which is based on machine learning techniques. In Section 4, we give the details of each machine learning step. In Section 5, we present our experiments in different summarization tasks and different levels of evaluation, and then we discuss the obtained results.

2. Previous Work

In this section, we describe the principal related works that deal with the evaluation of the content and the linguistic quality of a text summary.

2.1. Content Evaluation

The summary evaluation task started with the manual comparison of peer summaries with reference summaries. Achieving this task was an arduous and costly process. One of the first and famous tools of manual summary evaluation is SEE (Summary Evaluation Environment) [4]. It allowed human judges to manually evaluate the content and the linguistic quality (i.e. grammaticality, cohesion, coherence, etc.) of a summary. To evaluate content, human judges were used to compare a candidate summary (system summary) to an ideal summary. After that, [5] proposed PYRAMID which is a manual metric based on identifying the common ideas between a candidate summary and one or several reference summaries. These ideas are represented as semantic information units called Semantic Content Units (SCUs). The PYRAMID metric was used by the TAC and DUC conference to evaluate the content of candidate summaries. Several automatic metrics have been presented to treat the cost / time problem imposed by manual metrics. One of the well-known metrics in automatic text evaluation is ROUGE [6]. It measures the number of overlapping units between a candidate summary and reference summaries. There are many variants of the ROUGE metric which change according to the chosen unit of comparison such as *n*-gram (ROUGE-N), word sequences (ROUGE-L, ROUGE-W), and word pairs (ROUGE-S) between the candidate summary and the reference summaries. Afterwards, [7] proposed a new metric called BE (Basic Elements) which operates at the semantic level rather than the shallow or surface level like the ROUGE metric. To decompose each sentence in a summary to minimum semantic units called Basic Elements (BE), each summary requires a deep analysis (a semantic analysis). The final score relies on the overlap of BE units between a candidate summary and reference summaries. Later, Giannakopoulos *et al.* [8] introduced the AutoSummENG metric, which is based on the statistical extraction of textual information from the summary. The information extracted from the summary represents a set of relations between the summary's *n*-grams. A graph is constructed including the full set of relations and additional information concerning these relations. The estimation of the similarity degree is performed by comparing the graph of the candidate summary with the graph of each reference summary. Finally, the average similarity

degree between the candidate summary and all reference summaries is considered as the overall score of the candidate summary. In a subsequent work, Giannakopoulos and Vangelis [9] presented the Merge Model Graph (MeMoG) which is another variation of the AutoSummENG based on *n*-gram graphs. This variation calculates the merged graph of all the reference summaries. Then, it computes the similarity degree between the candidate summary graph and the merged graph of the reference summaries.

In a recent work, [10] developed the SIMetrix measurement; it assesses a candidate summary by comparing it with the source documents instead of reference summaries. The SIMetrix is a full automatic metric which does not depend on reference summaries. [10] computed ten measures of similarity based on the comparison between the source documents and the candidate summary. Among the used similarity measures we cite the cosine similarity, the divergence of Jensen-Shannon, the divergence of Kullback-Leibler, etc. In a more recent work, [11] developed the SERA (Summarization Evaluation by Relevance Analysis) metric, which is designed to evaluate scientific articles. This metric relies on the relevant content shared by a candidate summary and reference summaries. [11] used an information-retrieval-based method which treats summaries as search queries and then measured the overlaps of the retrieved results. A larger number of overlaps between the candidate summary and the reference summary indicates that the candidate summary has a higher content quality.

The observation of all of the previous cited metrics shows that each metric uses only one level of comparison (the lexical level, the syntactic level, the semantic level, etc.), while the combination of many comparison levels may overcome the limits of each metric. In addition, combining scores that rely on the comparison between candidate and reference summaries and scores that are based on comparing candidate summaries with source documents can overcome the limits of each type of comparison. For instance, it is important to compare between texts with similar lengths, but reference summaries cannot always cover all the formulations of important ideas presented in source documents. Nevertheless, the comparison between texts that have a big difference in terms of length remains a difficult task.

2.2. Linguistic Quality Evaluation

The language quality is an important factor in assessing the quality of a summary. Indeed, a good linguistic quality makes a summary easy to read and understand. In fact, in the TAC conference, the linguistic quality was based on the combination of five aspects, namely structure and coherence, grammaticality, non-redundancy, referential clarity and focus. During the DUC and the TAC conferences, the linguistic quality of a summary was evaluated by human judges that took into account the five linguistic aspects without using reference summaries or source documents. Accordingly, the judges did not take into account the relationship between the summary and the source documents and were expected to assess the summary as a separate document.

Because of the difficulty of manual evaluation, more work has been done in this area for automation. In this context, [12] evaluated mainly the local coherence of the summary using an entity grid model that captures the transitions of entities between two adjacent sentences. In this model, the text is represented as a matrix where each column contains one entity and each line contains a sentence. Each cell corresponds to the grammatical role of the entity in the sentence. The proposed method calculates the local coherence of the summary using the probability distribution of the entity's transitions. Many other researches like [13] and [14], explored the entity grid model to evaluate local coherence. In addition, [15] dealt with the assessment of grammaticality and coherence in the summary. They proposed to apply machine learning techniques to train a language model by referring to a corpus of manual summaries with parts of speech and /or chunk labels. After that, the learning model estimates the probability of grammatical acceptability of a sentence. To evaluate the structure and the coherence of a summary, [15] built a lexical chain which is spread over the entire summary to represent the sequences of related words. The lexical chain which is produced can provide information on the focus of each sentence, which in its turn contributes to the focus of the summary. Besides, [16] attempted to predict each of the five linguistic aspects mentioned previously. They identified several linguistic features that were grouped into classes. Then, they tried to identify the best

class of features for each linguistic quality aspect. Next, for each aspect, they built a model from each class of features. Finally, they built a meta-ranker for each aspect by combining the predicted scores from each model related to this aspect. Also, [3] evaluated summaries by constructing predictive models for overall responsiveness, PYRAMID and linguistic quality using a combination of content scores based on bi-grams and others related to linguistic quality. To build the predictive model, [3] tested three regression methods, namely canonical correlation, the Robust Least Squares, and the Non -Negative Least Squares. On the other hand, the CREMER metric [17] combined a content metric named TESLA-S [17] and a linguistic quality metric called "DICOMER" [17] to predict overall responsiveness. Some works have tried to predict overall responsiveness scores using the combination of content scores and linguistic quality features, but no one has combined them to predict linguistic quality score.

3. Our Method

Our method presents an alternative view of the problem of text summary evaluation through the use of a machine learning technique. It is based on the combination of several content scores and linguistic quality features to predict manual scores and more precisely PYRAMID scores and linguistic quality scores. The choice of the prediction of these two scores is stimulated by their reputation and their availability in the manual evaluations of the DUC and TAC evaluation conferences. In fact, PYRAMID is a manual score that reflects the coverage of important ideas present in the reference summaries by comparing the extracted SCUs from the candidate summary and the reference summaries. But, human judges cannot detect SCUs in a candidate summary with a poor linguistic quality; this is why we should include some linguistic features in the model that will predict PYRAMID scores. In addition, many studies ([2], [3], etc.) have proved that the combination of automatic content scores improves the correlation between the PYRAMID scores and the combination of automatic content scores.

For the linguistic quality, we are interested in providing a score that takes into account five linguistic aspects (namely structure and coherence, non-redundancy, focus, referential clarity and grammaticality) like the manual linguistic quality score used by the TAC conference which includes all five aspects. So, it is impossible to handle linguistic quality evaluation using one single score without combining several features that cover all linguistic quality aspects. In the features already used we have included some content features based on the overlap between extended textual units (bi-grams, tri-grams, etc.). In fact, those extended units can capture some grammatical and structural phenomena, whether the comparison is made with model summaries or source documents.

Consequently, to predict the content and the linguistic quality using a machine learning technique, we should go through the different phases presented in Figure 1.

As shown in Figure 1, the first phase of our method is the machine learning phase where we build the predictive model for each manual score. This phase consists of three steps: feature extraction, selection of relevant features and training and validation of the predictive model. All of those three steps will be described subsequently. The second phase is the exploitation phase where we will apply the model to predict a manual score (PYRAMID or linguistic quality) of the new candidate summary (unseen summary).

The use of our method will increase the correlation of the predicted scores with the manual scores. This is because the use of a single score or feature could not take into account all the aspects of content or of linguistic quality score, while our predictive model will consider more aspects that are present in the manual scores.

In the next section, we will detail the machine learning phase, which represents the basic substrate of the proposed method.

4. Machine Learning Phase

4.1. Feature Extraction

This first step computes all the feature values related to each candidate summary. In this step, we need several natural language processing tools such as the Stanford parser [18], the

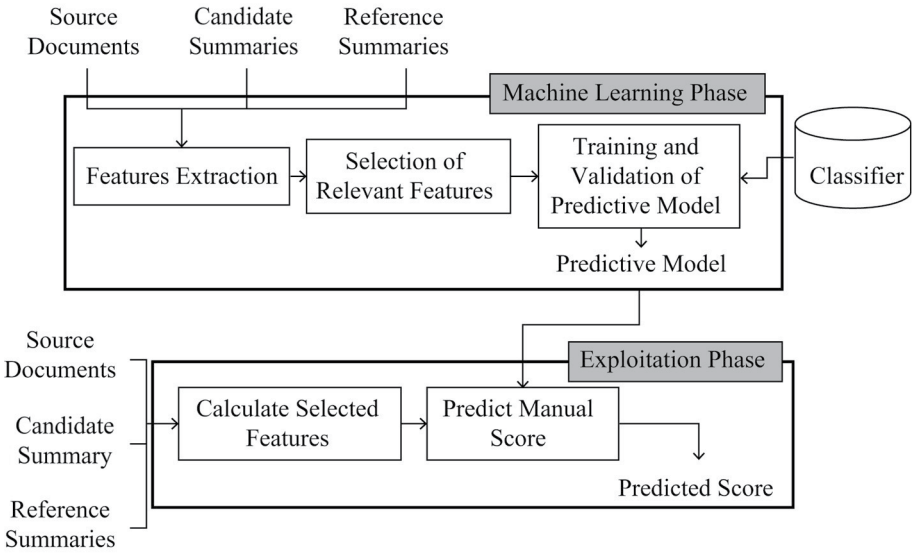

Figure 1. Method phases.

Stanford Tagger [19], the Stanford NER [20], the Stanford Coref [21], the srilm toolkit [22], etc., to calculate some linguistic quality features. However, the content features are based on many content metrics. Those content features require the use of reference summaries or source documents.

The goal of this phase is to transform the text summary input into a -feature matrix. In this phase, we use some new features and other features that are successfully used either in the assessment of readability or of content. For the linguistic features that have been used, we have tried to cover many linguistic aspects (e.g. grammaticality, non-redundancy, Structure and coherence, etc). In this work, we have included all the classes of features that were used in [23] and [24] (traditional readability measures, shallow features, part of speech features, etc.) and we have added some new features to existent classes and also other new classes of features. In the following subsections we will present the features related to each class.

4.1.1. ROUGE / BE Scores

We used ROUGE-N (R-N) scores based on the overlap of *n*-grams between the candidate summary and the reference summaries (such as ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-4). In addition to that, we included ROUGE-SU4 (R-SU4) based on skip-bigrams and uni-grams, ROUGE-L (R-L) based on the Longest Common Subsequence of *n*-grams and ROUGE-W (R-W) based on the Weighted Longest Common Subsequence of *n*-grams. Finally, we used a BE score based on the semantic units called BE (Basic Elements).

4.1.2. AutoSummENG Scores

We determined four variants of AutoSummENG. The first is AutoSummENG_W123 where we calculated AutoSummENG which includes *n*-grams of words of a length varying between [1,...,2] and which has a window of size 3 between *n*-grams. The second is AutoSummENG_W333 which uses tri-grams of words and which has a window of size 3 between *n*-grams. The third is AutoSummENG_C123 using *n*-grams of characters of a length between [1,...,2] and having a window of size 3. The fourth is AutoSummENG_W253 using *n*-grams of words of length between [2,...,5] and having a window of size 3.

4.1.3. Adapted ROUGE Scores

We maintained the adapted ROUGE scores that have been introduced in [24]: R-N_{Ad} which represents an adapted ROUGE score based on *n*-grams where *N* is a number between [2,...,5],

$R\text{-}L_{Ad}$ which represents a ROUGE adapted score based on the Longest Common Subsequence of *n*-grams, $R\text{-}S4_{Ad}$ which designs a ROUGE adapted score based on skip-bigrams, $R\text{-}W_{Ad}$ which designs a ROUGE adapted score based on the Weighted Longest Common Subsequence of *n*-grams.

4.1.4. SIMetrix Score Features

The SIMetrix metric involves different similarity measures such as cosine similarity, Kullback-Leibler (KL) divergence, Jensen-Shannon divergence, etc., for comparing the content of the source documents and that of the candidate summary. Because of the variation in the ways of similarity estimation between the different variants of SIMetrix [10], we used all the ten scores calculated by SIMetrix. Those scores are: the Kullback-Leibler divergence [25] between the source documents and the candidate summary (KLInputSummary); the KL divergence between the candidate summary and the source documents (KLSummaryInput); the unsmoothed version of Jensen-Shannon divergence [26] between the source documents and the candidate summary (unsmoothedJSD) and the smoothed one (smoothedJSD); the cosine similarity between the source documents and the candidate summary (cosineAllWords); the percentage of the descriptive words of the source documents that appear in the candidate summary (percentTopicTokens); the percentage of the candidate summary composed of the most descriptive words from the source documents (fractionTopicWords); the cosine similarity between the candidate summary and the most descriptive words in the source documents (topicWordOverlap); the probability of uni-grams of the summary given in the source documents (unigramProb); the multinomial probability of the summary given in the source documents (multinomialProb).

4.1.5. Traditional Readability Measure Features

The class of traditional readability measures we used includes six readability measures: Flesch-Kincaid Index (Ind.), Flesch Reading Ease, Automated Readability Index, Gunning Fog Index, Dale-Chall grade and SMOG.

4.1.6. Shallow Features

For shallow features, we adopted multiple features used by [23] and [24]: the average number of syllables per word (AvgSyllW), the average number of characters per word (AvgCharW), the average number of words per sentence (AvgWSent), the ratio between the candidate summary's size and the maximum size allowed by the TAC campaign (RatioW_MaxW), the logarithm of the number of sentences (logS), the logarithm of the number of characters (logC), the logarithm of the number of words (logW). In addition, we added a set of features based on lexical diversity which counts how many different words are used in a text. In fact, a high score of these features can ensure that the sentences of a summary are less repetitive and have a rich vocabulary. Those features are: the number of distinct words, the density of distinct words, the root of the density of distinct words, the correct density of distinct words (DensCorrDistWord), the bi-logarithmic density of distinct words, the Uber Index (Uber_Index). In addition, we determined for each candidate summary two features based on paragraph length: the average number of sentences per paragraph, the average number of words per paragraph. In fact, a short paragraph can be more easily understood and can have fewer problems of co-referencing and of liaison between the ideas of its sentences. Moreover, we calculated the density of stop words (DensStopW).

4.1.7. Language Modeling Features

For language modeling features we calculated the nine following features for each candidate summary: the Log probability of uni-grams (logProbUnigram), the measure of perplexity for unigrams normalized by the total number of uni-grams (pplUnigram), the measure of perplexity for uni-grams with exclusion of the sentence end tags (ppl1Unigram), the Log probability of bi-grams (logProbBigram), the measure of perplexity for bi-grams normalized by the total number of bi-grams (pplBigram), the measure of perplexity for bigrams with exclusion of the sentence end tags (ppl1Bigram), the Log probability of tri-grams (logProbTrigram), the measure of perplexity for tri-grams normalized by the total number of tri-grams (pplTrigram),

the measure of perplexity for tri-grams with exclusion of the sentence end tags (ppl1Trigram).

4.1.8. Part-of-Speech Features

Our Part-of-speech (POS) features are based on categorical word frequencies. We calculated the number (Nb), the average (Avg) and the density (Dens) of each of the following functional and content words: determinants (DET), coordinating conjunctions (CC), prepositions and subordinating conjunctions (PSC), personal pronouns (PRP), nouns (N), verbs (V), adjectives (ADJ), adverbs (ADV). We added some part-of-speech features which are related to nouns and verbs which are the most important and essential part of content words for a text summary. This is because a summary must contain fewer description details (i.e., fewer adjectives and adverbs) and more important actions expressed by nouns and verbs. The added features which are calculated for a candidate summary are: the density of verbs and nouns, the ratio between the number of nouns and the number of verbs, the average number of nouns and verbs, the ratio between the number of nouns and verbs and the number of adjectives and adverbs, the ratio between the number of each type of verb (infinitive, imperative, participle, model) and the total number of verbs. Finally, we added the number, the average and the density of lexical word (LexW) (nouns, verbs, adjectives and adverbs).

4.1.9. Syntactic Features

For the syntactic features, we adopted a range of parse tree based features used by [23] and [24]. Those features include the number and the average number of: noun phrases (NP), verb phrases (VP), prepositional phrases (PP), clauses (SBAR). Furthermore, we maintained the average height of the parse tree and the average number of dependency relations.

4.1.10. Named Entity Based Features

For each candidate summary, we implemented the following three named-entity features used in [23] and [24]: the number of named entities (Ent), the density of named entities and the average of named entities. In addition, we counted the number of distinct entities, the density of distinct entities, the average number of distinct entities and the entities' diversity. The latter feature is equal to the ratio between the number of distinct entities and the total number of entities.

4.1.11. Coherence Features

We will focus first on local coherence. In fact, local coherence gauges the continuity of ideas between adjacent sentences. Therefore, we estimated this continuity using several similarity and distance measures. For each candidate summary, we computed the average similarity or distance between adjacent sentences using: the Levenshtein distance, the cosine similarity, the Jaccard distance, the divergence of Jensen-Shannon, the Kullback-Leibler divergence, the Pearson correlation, the Dice index and the overlap coefficient.

Second we evaluated the coherence of the text summary using features based on discourse relations. In fact, a coherent text can be represented as a combination of text units connected by discourse relations. Discourse relations such as cause, contrast or elaboration are important to understand the relation of each sentence in the text to the others; this forms a coherent text. In our work, we calculated for each candidate summary: the number of discourse relations (NbDisc), the average number of discourse relations per sentence (AvgDisc) and the density of discourse relations (DensDisc).

4.1.12. Co-Reference Features

We used the Stanford Coref [21] to allow us identify the different co-reference relations in a summary and the sentences where the co-reference and its antecedent are found. From those pieces of information, we extracted the number of times a pronoun has no antecedent (CorefWithoutAnt), the number of times a pronoun has an antecedent (corefWithAnt), whether its antecedent is in the current sentence (AntSameS), in the previous sentence (AntPrevS) or not in the same sentence or in the previous sentence (AntOther). In addition, we determined the ratio between the number of co-references without antecedent and the total number of co-references with an antecedent (RatWithAntWithoutAnt)

and vice versa (RatWithoutAntWithAnt), the number of pronouns without antecedent to the total number of words (RatWithoutAntNbW) and the number of pronouns without antecedent to the total number of pronouns (RatWithoutAntNbPron).

4.1.13. Redundancy Features

To calculate these features, we compared each sentence in the candidate summary with the other sentences by using a lexical similarity measure. Four similarity measures were adopted, namely the cosine similarity, the Dice coefficient, the overlap coefficient and the Jaccard index. For each similarity measure, we determined the maximum redundancy (e.g. the average of maximum similarities between each sentence and other sentences in the summary) and the average redundancy (average similarity between sentences) between sentences.

4.1.14. Specific-General Sentence Features

In fact, a good summary requires generalizations. Therefore, it should contain the maximum number of general sentences. For this reason, we decided to include features that tell us about the types of sentences used in a candidate summary. To calculate those features, we used the speciteller tool [27] which gives for each sentence a score (specific-general score) between 0 and 1 where 0 is the score for the most general sentences and 1 is the score for the most specific ones. For each summary, we determined the ratio between the sum of the specific-general scores of all sentences and the number of sentences in a summary (RatSpecGenScoreNbSent). In addition, we identified the ratio between the number of sentences with a score > 0.5 and the number of sentences with a score < 0.5 (RatGenSentSpecSent).

4.2. Selection of Relevant Features

For each task and for each manual score (PYRAMID, linguistic quality), this step receives as input a matrix X of features and a vector V of values of one of the manual scores where x_{ij} is the value of the j^{th} feature for the i^{th} candidate summary, i is between 1,...,n and j is between 1,...,m. It should be noted that the candidate summaries in the learning phase can be system summaries or reference summaries. Indeed, in order to increase the size of the training corpus, reference summaries were included. This step allows us to select the most relevant features that must be kept for the training step and to remove unneeded, irrelevant and redundant features which may decrease the performance of the predictive model. So, usually, the use of all possible features does not give the best predictive model because of the presence of redundant and irrelevant features in the model. In addition, the absence of a feature selection step may, on the one hand, introduce bias into the built model which can lead to over-fitting, and on the other hand, induce a greater computational cost. In general, the selection of relevant features is as important as the choice of the learning algorithm. Similarly, it is important to say that the choice of the selection algorithm depends on the type of machine learning algorithm to use (regression or classification).

Several selection algorithms have been tested in the case of predicting the content score and the linguistic quality score. To select the relevant features for the content score model, we used a wrapper method that has given the best predictive model. This method assesses subsets of features according to their usefulness to a given classifier. According to [28], the Wrapper subset evaluator [29] selection methods are generally considered better than the filtering ones. However, to select relevant features for the linguistic quality model, we used the GainRatioAttributeEval or OneRAttributeEval method, which has given the most appropriate features to build the predictive model. The two methods represent two filter methods which select subsets of features as a pre-processing step, independently of the classifier chosen in the training phase.

4.3. Training and Validation of the Predictive Model

The last step in the machine learning phase receives as input a matrix of size (n, k) where k is the number of features considered pertinent by the previous step and a vector containing the manual scores. This step helps to build and validate the predictive models of the two manual scores: the PYRAMID score and the linguistic

quality score. The PYRAMID score represents a continuous value between 0 and 1. For this reason, we built the predictive model of this score using a linear regression technique. To build the best predictive model, we tested many linear regression algorithms implemented by the Weka environment [30] such as GaussianProcesses, LinearRegression, LeastMedSq, RandomForest, etc. The linguistic quality score represents a discrete number between 1 and 5; this is why we chose a supervised classification to predict the linguistic quality of a summary. To make the predictive model of the linguistic quality score, the MultilayerPerceptron, the NeuralNetwork, the SMO, the LWL, or the Dl4jMlpClassifier (a Deep learning classifier) can be used.

Moreover, we tried to produce models for content and for linguistic quality scores by using ensemble learning which usually promotes the production of more accurate solutions than a single learning algorithm. In our experiment we used three ensemble learning algorithms which are implemented in the Weka environment: Bagging [31], Vote [32], AdditiveRegression and Stacking [33].

In our method, after testing the algorithms used for each type of score, we adopt the one that produces the best predictive model. The validation of each model is performed by a cross-validation method with 10 folds. We divide the dataset into 10 folds and we repeat the experiment 10 times. Each time, we preserve one fold for the test and the rest for the training of the model. Finally, for the linguistic quality, we report the accuracy (it represents the proportion of instances that were classified correctly) and the kappa coefficient calculated from the 10 times. For the content score, we determine the Pearson correlation [34] and the Root Mean Square Error RMSE. The RMSE can be interpreted as the average deviation of the manual score between the predicted and actual values. This measure penalizes models which make big prediction errors compared to the manual scores (the readability score or the PYRAMID score).

5. Experimentation

We tested our method for summary level evaluation in initial summary task (Task A) [35] and update summary task (Task B) [35] by trying to predict PYRAMID and linguistic quality scores. On the system level, for each task, we will just average the predicted scores of all the candidate summaries of each system.

5.1. Data Set

The Data Set used in the study is taken from the updated summary task of the TAC 2008 [35] conference. This data set consists of the source documents, the manual summaries (reference summaries) and the system summaries. This task consists of summarizing a set of documents (A) which deals with a particular event and then of summarizing a set (B) which addresses the evolution of the same event and considers the knowledge of the set (A). This corpus includes 5568 system summaries that are automatically generated by the 58 participating systems where each system produced 96 summaries: 48 summaries (48 is the number of collections of source documents) for each set of documents (A and B). The corpus also includes 384 (96∗4) reference summaries (4 reference summaries for each collection in a set of documents). Thus, each system summary can be assessed by comparing the four reference summaries. Similarly, a reference summary can be evaluated by comparing it with the other three model summaries. In our experiments in summary level evaluation, each model is produced using 2976 candidate summaries (coming from all collections) where 2784 are system summaries and 192 are reference summaries.

5.2. Content Score Evaluation

5.2.1. Summary Level

In this subsection, we begin by citing in Table 1 the selected features for the content score prediction in initial summary task. In Table 1, we notice the selection of multiple content scores in addition to many linguistic quality features. We have observed the presence of features related to reference clarity and redundancy (AntPrevSent, RedondAVGdice). This means that when evaluating the content, we need to have a candidate summary with clear reference resolution and without redundancy. In addition, we notice the presence of language modeling and

syntactic features which can be indicators of the fluency and the grammaticality of a summary.

Now, we give in Table 2 the list of selected features (feat.) used in the update summary task. From this table, we notice that in the update summary task too, multiple content scores (sc.) are selected as relevant features. In addition, many linguistic quality features are selected as relevant ones. Besides, the importance and the necessity of including linguistic quality features is clearly shown through the use of features related to diverse aspects of linguistic quality like referential clarity, non-redundancy, local coherence, grammaticality (i.e. AvgNPSent, AvgPPSent, RatioWordMaxWord which can be an indicator of the truncation of the last sentence in the summary).

Table 1. List of selected features to predict content score in summary level for initial summary task.

Features	Initial Summary
Content features	R-1, R-2, R-3, R-4, R-SU4, R-W, AutoSummENG_W333, AutoSummENG_W123, AutoSummENG_W253, KLInputSummary, KLSummaryInput, unsmoothedJSD, smoothedJSD, cosineAllWords, percentTopicTokens, fractionTopicWords, topicWordOverlap, unigramProb, multinomialProb
Linguistic quality features	NB_DET, NB_PSC, DENS_DET, DENS_N, DENS_V_N, Uber_Index, AvgSBARSent, NB_PP, NB_SBAR, AVG_Height_ParseTree, AVG_NB_dep_sent, pplUnigram, pplBigram, NB_Ent, AvgKLdiv, AntPrevSent, RatWithoutAntNbWord, RedondAVGdice

Table 2. List of selected features to predict content score on summary level for update summary.

Features	Update Summary
Content features	R-1, R-2, R-4, R-SU4, R-L, R-W, $R\text{-}4_{Ad}$, $R\text{-}5_{Ad}$, $R\text{-}S4_{Ad}$, BE, AutoSummeng_C333, AutoSummENG_W123, autosummeng_W253, KLInputSummary, unsmoothedJSD, smoothedJSD, cosineAllWords, percentTopicTokens, topicWordOverlap, unigramProb, multinomialProb
Linguistic quality features	NNB_DET, NB_CC, AVG_DET, AVG_SC, AVG_N, DENS_PSC, DENS_PRP, Ratio_ImpV_V, AVG_LexW, fleschK, GunFog, DaleChall, AvgSyllW, AvgCharW, AvgWord_Sent, AVG_Sent_Parag, DensCorrDistWord, logS, logW, AvgNPSent, Avg_PPSent, NB_PP, logProbUnigram, logProbBigram, ppl1Bigram, logProbTrigram, pplTrigram, pl1Trigram, Nb_Ent, AVG_Ent_Sent, AVGLevenDist, AVGJacSim, AVGJSDiver, AntPrevSent, AntSameSent, AntOtherSent, NB_Disc, AVG_Disc

We examine the usefulness of the selected features in the prediction of the content score. We use the following single classifiers: GaussianProcesses, LinearRegression, LeastMedSq, MultilayerPerceptron and RandomForest from the Weka environment to train the model. Then, we validate the model using a 10-fold cross-validation. The correlation and the RMSE generated by each classifier are presented in Table 3. This table shows the performance of the selected features in building models using several single and ensemble classifiers in the initial and update summary tasks. In the initial summary task (Task A), the results show that the model built from the ensemble learning classifier Stacking produced the best correlation (0.7906) and the lowest RMSE (0.1133).

In addition, the results obtained through all the ensemble learning classifiers were, in general, better than single classifiers. The difference be-

Table 3. Pearson correlation with PYRAMID and RMSE (between parentheses) for both tasks using the selected features.

Classifiers	Task A	Task B
Single classifiers		
GaussianProcesses	0.7690(0.1185)	0.8203(0.1097)
LinearRegression	0.7530(0.1218)	0.7648(0.1228)
LeastMedSq	0.7496(0.1226)	0.7424(0.1295)
SMOReg	0.7661(0.1201)	0.8275(0.1072)
MultilayerPerceptron	0.6948(0.1361)	0.7047(0.1473)
RandomForest	0.7766(0.1173)	0.8196(0.1102)
Ensemble learning classifiers		
Additive regression	0.7881(0.1142)	0.8300(0.1068)
Vote	0.7904(0.1135)	**0.8342(0.1061)**
Bagging	0.7723(0.1180)	0.8191(0.1101)
Stacking	**0.7906(0.1133)**	0.8337(0.1052)

tween the best and the least performing classifier in terms of correlation is equal to 0.0958. This presents a significant difference for the task of initial summary evaluation.

On the update summary level, Table 3 indicates that SMOReg is the best single classifier in predicting the PYRAMID score with a correlation of 0.8275 and an RMSE of 0.1072. The best ensemble learning classifier is Vote which provides a model having a correlation of 0.8342 and an RMSE of 0.1061. Another notable observation is that the correlation in the update summary task is more important than the one in the initial summary task. Furthermore, the RMSE has less important values in the update summary task than in the initial summary task. This indicates that the deviation between the predicted and the actual values is less important in the update summary task than in the initial summary task.

After obtaining the best performing classifier for building the predictive model of the PYRAMID score, on both tasks, we move on to the comparison between the performance of the best model obtained by combining selected features (feat.) and the baseline metrics such as ROUGE-2, ROUGE-SU4 and BE that were adopted by the TAC conference as baseline metrics, the model combining content scores (sc.) and the model combining all features. Table 4 details the different correlations and RMSEs of baseline metrics and our experiments. It should be noted that, in the initial summary task, the models of our experiments are built using the Stacking ensemble learning classifier. In addition, it should be noted that in the update summary task, the models of our experiments are made using the Vote ensemble learning which integrate two classifiers RandomForest and SMOReg. From Table 4 and in both tasks, we see the gap between the baseline metrics and our experiments, regardless of whether we used the selected features, just content scores, or all features. Moreover, we notice that the inclusion of linguistic quality features in the best model produced improves the performance of this model compared to the model containing just content scores.

In addition, we note that the inclusion of all features in one model does not give the best predictive model which justified the selection of relevant features.

Table 4. Pearson correlation with PYRAMID score and RMSE (between parentheses) in both tasks of summary level evaluation.

Scores	Task A	Task B
Baselines		
ROUGE-2	0.5990(0.1482)	0.5825(0.1549)
ROUGE-SU4	0.5090(0.1399)	0.6200(0.1495)
BE	0.4493(0.1653)	0.5534(0.1588)
Our experimentations		
Combining content sc.	0.7763(0.1167)	0.8165(0.1248)
Combining selected feat.	**0.7906(0.1133)**	**0.8342(0.1061)**
Combining all feat.	0.7797(0.1159)	0.8206(0.1101)

5.2.2. System Level

In system level evaluation, we estimate the quality of a summarization system; in other words, the system assessment is done by taking into account the quality of all the summaries that are produced by this system. In this article, we tried to calculate the quality of a system by determining the average of the predicted scores for the summaries produced by the same system. The formula for calculating the performance of the content or the linguistic quality of a summarization system is as follows:

$$Score_{system} = \frac{\sum_{i=1}^{N} Score_{sum_i}}{N} \quad (1)$$

where N is the number of summaries produced by a definite system and $Score_{sum_i}$ is the predicted score of the summary i. To evaluate this method of calculating the content score for a system, we study the correlation of Pearson (P), Spearman (S) [36] and Kendall (K) [37] with the PYRAMID score and the Scoresystem score. Table 5 details the different correlations between the PYRAMID score and the $Score_{system}$ score or the baseline scores.

Table 5 shows how good each baseline is when the $Score_{system}$ is compared to the PYRAMID score. As can be seen in this table, the best result is obtained by our $Score_{system}$. It has the best correlation with the PYRAMID score in both

Table 5. Pearson, Spearman and Kendall correlations with PYRAMID score in both tasks in system level evaluation.

	P	S	K	P	S	K
Scores	**Initial Summary**			**Update Summary**		
R-2	0.8718	0.9364	0.8050	0.9009	0.9588	0.8322
R-SU4	0.8741	0.9007	0.7477	0.8458	0.9323	0.7796
BE	0.9188	0.9329	0.7889	0.9188	0.9560	0.8297
$\boldsymbol{Score_{system}}$	**0.9805**	**0.9781**	**0.8825**	**0.9997**	**0.9987**	**0.9820**

tasks and with the three types of correlation measures. With all correlation measures, there is an important difference between the baselines and our system score ($Score_{system}$).

5.3. Linguistic Quality

We evaluated the linguistic quality of a summary in the initial and the update tasks. To get better results we tried to test many single and ensemble learning classifiers. Then we compared the best model produced with four traditional measures of readability (baselines) and with a model generated using only linguistic quality features. The four traditional measures of readability are the Gunning Fog Index [38], the Flesch Reading Ease [39], the Flesch-Kincaid Index [40] and the Automated Readability Index [41]. Those measures are used to assess text readability at the school grade level of the reader. But, because on the one hand, there is a lack of measures that can be used as baselines, and on the other hand, the linguistic quality and the readability at the school grade level have multiple common points (like fluency, structure, non-redundancy etc.), we decided to use those measures as baselines.

5.3.1. Summary Level

To build a model that predicts linguistic quality in the initial task, we used the selected features cited in Table 6. From Table 6, we note that the selected features come from all classes of features. This indicates that each class of features covers some linguistic quality aspects. In addition, we notice that many content features based on *n*-grams overlap are used (i.e., ROUGE-2, ROUGE-3, etc.). Furthermore, all of the adapted ROUGE variants are used. It should be noted that adapted ROUGE variants are designed to evaluate the grammaticality and the structure of a candidate summary [24].

Table 6. List of selected features to predict linguistic quality score in summary level for initial summary task.

Features	Initial Summary
Content features	R-1, R-2, R-3, R-4, R-SU4, R-L, R-W, $R\text{-}2_{Ad}$, $R\text{-}3_{Ad}$, $R\text{-}4_{Ad}$, $R\text{-}5_{Ad}$, $R\text{-}L_{Ad}$, $R\text{-}W_{Ad}$, $R\text{-}S4_{Ad}$, BE, AutosummENG_C333, AutoSummENG_W333, KLSummaryInput, AutoSummENG_W123, KLInputSummary, AutoSummENG_W253, unsmoothedJSD, smoothedJSD, cosineAllWords, percentTopicTokens, fractionTopicWords, topicWordOverlap, unigramProb, multinomialProb
Linguistic quality features	NbDET, NbCC NbPSC, NbPRP, NbADJ, NbV, NbN, NbADV, AvgDET, AvgCC, AvgPSC, AvgPRP, AvgADJ, AvgV, AvgN, AvgADV, DensDET, DensCC, densPSC, DensPRP, DensADJ , DensV, DensN, DensADV, DensV_N, ratio_N_V, Avg_N_V, DensStopW, fleschK, readInd, AvgSyllW, AvgCharW, AvgWSent, AvgWParag, RatioW_MaxW, logS, AvgNP, AvgVP, AvgPP, AvgSBAR, NbNP, NbVP, NbPP, NbSBAR, AVG_Height_ParseTree, AVG_NB_dep_Sent, logProbUnigram, pplUnigram, ppl1Unigram, logProbBigram, pplBigram, ppl1Bigram, logProbTrigram, pplTrigram, ppl1Trigram, NbEnt, DensEnt, AvgEnt, AVGLevenDist, AVGCosSim, AVGJacSim, AVGJSDiver, AVGPearCor, AVGdice, AVGoverlap, CorefWithoutAnt, corefWithAnt, AntSameS, AntPrevS, AntOther, RatWithAntWithoutAnt, RatWithoutAntWithAnt, RatWithoutAntNbW, RedondMaxDice, RedondAVGdice, RedondMaxOver, RedondAVGover, RedondMaxJacc, RedondAVGjacc, redondMaxCos, RedondAVGcos

Table 7. List of selected features to predict linguistic quality score in summary level for update summary task.

Features	Update Summary
Content features	R-1, R-2, R-3, R-4, R-SU4, R-L, R-W, $R\text{-}2_{Ad}$, $R\text{-}3_{Ad}$, $R\text{-}4_{Ad}$, $R\text{-}5_{Ad}$, $R\text{-}L_{Ad}$, $R\text{-}S4_{Ad}$, BE, AutoSummeng_C333, AutoSummENG_W123, KLInputSummary, Autosummeng_W333, Autosummeng_W253, KLSummaryInput, unsmoothedJSD, cosineAllWords, unigramProb
Linguistic quality features	NbCC, NbPSC, NbPRP, NbADJ, NbV, NbN, AvgDET, AvgCC, AvgPSC, AvgPRP, AvgADJ, AvgV, AvgN, AvgADV, DensDET, DensCC, densPSC, DensPRP, DensADJ, DensV, densN, DensADV, DensV_N, ratioN_V, avgN_V, RatioInfV_V, RatioImpV_V, RatioPartV_V, RatioModV_V, NbLexW, DensLexW, AvgLexW, fleschK, fog, DaleChall, SMOG, AvgSyllW, AvgWSent, AvgSentParag, AvgWParag, RatioW_MaxW, NbDistinctW, TTR, Root_TTR logS, logC, logW, AvgNP, AvgVP, AvgPP, AVG_Height_ParseTree, AVG_NB_dep_Sent, logProbBigram, pplBigram, NbEnt, DensEnt, NbDistEnt, DensDistEnt, AvgDistEnt, EntDiver, AVGLevenDist, AVGCosSim, AVGJacSim, AVGJSDiver, AVGKLDiver, AVGdice, AVGoverlap, CorefWithoutAnt, corefWithAnt, AntSameS, AntPrevS, AntOther, RatWithAntWithoutAnt, RatWithoutAntWithAnt, RedondMaxDice, RedondMaxOver, RedondAVGover, RedondMaxJacc, RedondAVGjacc, redondMaxCos, RedondAVGcos, NbDisc, AvgDisc, DensDisc

Table 7 shows the different features selected by the OneRAttributeEval method in update summary tasks. As in the initial summary task, Table 7 shows that selected features come from all classes of features.

Also, most of the adapted ROUGE variants are used. To study the predictive power of the selected features, we trained and experimented with five single classifiers and three ensemble learning classifiers, on initial and update summary tasks. The basic classifiers used are MultilayerPerceptron, SMO, NeuralNetwork, LWL (LocallyWeighted Learning) [42] and RandomForest. The classification accuracy and the kappa generated by these models in both tasks are presented in Table 8. We adopted mainly the accuracy because it has been used by most studies [12], [16] and [15] related to the evaluation of linguistic quality.

In general, by observing Table 8, we see low values of accuracy and of kappa with all classifiers without exception. This is not surprising because, firstly, the summary level evaluation has always been a challenging task [16] and [24]. Secondly, the assessment of the linguistic quality of a summary is a difficult task because it involves various linguistic aspects.

From Table 8, on the initial summary level we find that the SMO classifier generates the highest accuracy (51.0081%) and the best kappa (0.3178). In the update summary task, the best accuracy (49.2608%) is achieved by the RandomForest classifier, while, the best kappa (0.3032) is obtained by Vote classifier.

Table 8. Accuracy and kappa (between parentheses) for various classifiers using the selected features in both tasks.

Classifiers	Task A	Task B
Single classifiers		
MultilayerPerceptron	43.8844%(0.2409)	40.1882%(0.2095)
SMO	**51.0081%(0.3178)**	47.5806%(0.2890)
NeuralNetwork	47.3480%(0.2301)	44.1868%(0.2189)
LWL	49.328%(0.2793)	48.5215%(0.2912)
RandomForest	50.0336%(0.2873)	**49.2608%**(0.3005)
Ensemble learning classifiers		
Vote	48.7903%(0.2498)	48.7903%**(0.3032)**
Bagging	48.9247%(0.2600)	49.1263%(0.2918)
Stacking	49.0591%(0.2873)	46.5726%(0.2711)

After having obtained the best model (built by combining "Comb." selected features) in terms of accuracy, we moved on to comparing it with the traditional readability measures (e.g. Gunning Fog Index "Gun. Fog. Ind.", Automated Readability Index "Aut. Read. Ind."), the model built with only linguistic quality features (ling. Qual. Feat.) and the model built with all features. It should be noted that all experimentations, in the initial summary task, are performed using the SMO classifier, while in the update summary task, they are achieved using the RandomForest classifier. Table 9 shows the accuracy and the kappa of baselines and of our experiments.

In both tasks, Table 9 shows the difference between the accuracy of the baseline metrics and our experiments. For example, in the initial summary task, we observe the difference between the model built from the selected features using the SMO classifier and traditional measures that exceed 13%. In addition, it was noticed for all traditional readability measures that the value of kappa is close to zero. This indicates a very weak agreement between the values of the predicted score and the actual score values of the manual score. In addition, the combination of only linguistic quality features has an accuracy of 47.4462%; the decrease of accuracy indicates that the content features have an influence on the linguistic quality evaluation. The same phenomenon is encountered in the update summary; there is a difference between the combination of only linguistic quality features and the combination of selected features. Furthermore, we notice that the use of selected features instead of using all features increases the accuracy and the kappa, in both tasks.

A closer inspection of the current prediction models of linguistic quality reveals the restriction of the use of a single model to predict the linguistic quality of the summaries produced by different collections that handle different topics in the corpus. This indicates that the heterogeneity of the collections from the point of view of writing style, vocabularies used, structures used, length of sentences, of paragraphs and of texts, etc., can influence the performance of the predictive model. Since each collection has its own characteristics, the features and the algorithm that perform best will not be the same for all collections.

To improve the ability of the models to predict the correct score of each summary, we need to build, for each collection, a model that predicts the scores of summaries coming from the same collection. Therefore, for both initial and update summary tasks, we tried to build for each task 48 models for the 48 collections available in the corpus used.

For the initial summary task, the 48 models are constructed using one of the four single classifiers (i.e. SMO, NeuralNetwork, MultilayerPerceptron, LWL), while for the update summary task, the models built are constructed using one of the three single classifiers (i.e. SMO, Dl4jMlpClassifier, MultilayerPerceptron). In addition, the Bagging ensemble learning classifier is used in both tasks (it gives the best model in

Table 9. Accuracy and kappa (between parentheses) of linguistic quality models in both tasks of summary level evaluatuion.

Scores	Task A	Task B
Baselines		
Gunning Fog Ind.	36.8952%(0.0002)	33.8710%(0.0000)
Flesch Reading Ease	36.8952%(-0.0002)	33.8038%(-0.0008)
Flesch-Kincaid Ind.	36.8616%(-0.0005)	33.8710%(0.0000)
Aut. Read. Ind.	36.8520%(0.0001)	33.9046%(0.0006)
Our experimentations		
Comb. ling. Qual. feat.	47.4462%(0.2408)	45.5981%(0.2587)
Combining selected feat.	**51.0081%(0.3178)**	**49.2608%**(0.3005)
Combining all feat.	49.3952%(0.2773)	47.6478%(0.2905)

some collections). We should note that the same proposed method is applied to each collection. Table 10 summarizes the results obtained with the construction of a model for each collection in both tasks.

Table 10 shows the improvement of the accuracy and kappa using one model per collection in both tasks. This Table shows that in initial summary the best accuracy among all the collections is that of collection 34 and is equal to 87.0968%. However, the best kappa is obtained with collection 18 and is equal to 0.6728. This shows the improvement obtained with regard to previous results where the accuracy did not go beyond 51.0081% and the kappa did not exceed 0.3178. Despite the remarkable improvement in the results, some collections have received low values in terms of accuracy and kappa. But, for accuracy, for instance, there are only two collections having less than 50% of accuracy. In addition, more than half of the collections have an accuracy which is greater than 60%. As presented in Table 10, in the update summary, the best accuracy is equal to 87.0968% and the best kappa obtained is 0.6416. Moreover, the average accuracy per collection is equal to 60.4167%, which is greater than the accuracy obtained by one predictive model for all the collections.

5.3.2. System Level

We used the same method (used for content evaluation) of averaging summary scores produced from the same summarization system.

The Pearson, the Spearman and the Kendall correlation of each baseline and of the $Score_{system}$ are shown in Table 11.

As can be seen from Table 11, in both tasks, our $Score_{system}$ performs better than the baseline measures (e.g. Flesch Reading Ease "Flesch-R. Ease", Flesch-Kincaid Index "Flesch-K. Ind."). Also the correlation between baselines and the linguistic quality measure is in general low with all correlation measures with the exception of the Pearson correlation, which gives in general a moderate correlation.

On the contrary, our $Score_{system}$ has in both tasks a very good Kendall correlation and nearly perfect Pearson and Spearman correlations.

6. Conclusion and Future Work

In this paper, we presented a method of content and linguistic quality evaluation for text summaries. Our work has been motivated by the lack of efficient and accurate automatic

Table 10. Recapitulation of the results obtained by applying the method of a model by collection on both tasks.

Initial summary task				
	Collection	**Classifier**	**Accuracy**	**Kappa**
Collection with the best accuracy	34	SMO	87.0968%	0.5595
Collection with the best kappa	18	SMO	80.6452%	0.6728
Collection with the lowest accuracy	27	Bagging	41.9355%	0.6728
Collection with the lowest kappa	37	LWL	50.0000%	0.1683
The average of all collections		LWL	63.8105%	0.3992
Update summary task				
	Collection	**Classifier**	**Accuracy**	**Kappa**
Collection with the best accuracy	32	MultilayerPerceptron	87.0968%	0.4233
Collection with the best kappa	46	Dl4jMlpClassifier	80.6452%	0.6416
Collection with the lowest accuracy	29	MultilayerPerceptron	46.7742%	0.2568
Collection with the lowest kappa	30	Dl4jMlpClassifier	59.6774%	0.2128
The average of all collections			60.4167%	0.3708

Table 11. Pearson, Spearman and Kendall correlations with linguistic quality score in both tasks of system level evaluation.

	P	S	K	P	S	K
Scores	**Initial Summary**			**Update Summary**		
Gun. Fog. Ind.	–0.4993	–0.2753	–0.1845	–0.4383	–0.1396	–0.0895
Flesch-R. Ease	0.2363	0.1779	0.1206	0.2010	0.1164	0.0754
Flesch-K. Ind.	–0.6546	–0.3616	–0.2474	–0.5995	–0.2567	–0.1747
Aut Read. Ind.	–0.7006	–0.3765	–0.2586	–0.6547	–0.2538	–0.1794
$\boldsymbol{Score_{system}}$	**0.9899**	**0.9859**	**0.9207**	**0.9994**	**0.9977**	**0.9788**

tools that evaluate the content and the linguistic quality of a summary. The proposed method is based on the construction of models that combine selected features which come from a large set of features that cover several linguistic aspects and several types of overlap between the candidate summary and reference summaries or source documents. For both scores, the combination of features is performed by testing many single and ensemble learning classifiers.

We have evaluated our method on two levels of granularity: the system level and the summary level and in two evaluation tasks: the initial summary task and the update summary task. On summary level, we have noticed that the model built using selected features that evaluate the content has the best correlation (0.7906 for the initial summary task) with the PYRAMID score. Furthermore, for linguistic quality evaluation and in both tasks, we also noted that the predictive model built using selected features has the best accuracy (51.0081% for the initial summary task) compared to baselines. In addition, we have built 48 models for 48 collections: a model for summaries from the same collection. We noticed that the accuracy of models has been increased for most collections: the best accuracy is 87.0968%, which is obtained with collection 34 in the initial summary task and collection 32 in the update summary task. This increase confirms our assumption that each collection has its specificity (writing style, sentence length, sentence complexity, etc.) since each one has a different topic.

In system level evaluation, for a specific task and a predicted score $Score_{system}$ (content or linguistic quality score), we calculated the average of the predicted score values of all the summaries that were built from the same summarization system. In both tasks, the average of the predicted content scores of each system $Score_{system}$ correlates best with the PYRAMID score. Likewise, the $Score_{system}$ correlates better with the manual linguistic quality score than with the baselines. In both tasks, it has been noted that there is a big gap between the correlation of the $Score_{system}$ with the manual linguistic quality score and between the correlation of the baselines with the manual linguistic quality score. Indeed, in both tasks and for both scores, our method has provided good performance compared to baselines. All the obtained results prove that, first, the combination of content and linguistic quality features to predict PYRAMID and linguistic quality scores can give better performance than single ones like content (ROUGE, BE, etc.) or linguistic quality (SMOG, FOG, etc.) scores. Second, we can affirm that adding linguistic features for the prediction of content score or content scores for the prediction of linguistic quality also improves the performance of the two prediction manual scores, PYRAMID and linguistic quality. Therefore, this means that there is a relation between the evaluation of content and the evaluation of linguistic quality. Third, we can assert that the selection of relevant features can on the one hand improve the performance of the predictive model and on the other hand provide faster and more cost-effective prediction models.

As perspective work for linguistic quality evaluation, we aim to study the causes that make the use of a classifier good in some collections and bad in the others. In addition, we want to study the reasons of the weakness of the accuracy and the kappa in some collections.

References

[1] A. Nenkova *et al.*, "Structural Features for Predicting the Linguistic Quality of Text – Applications to Machine Translation, Automatic Summarization and Human-Authored Text", *Proc. of EMNLP Conf.*, 2010, pp. 222–241.

[2] J. M. Conroy and H. T. Dang, "Mind the Gap: Dangers of Divorcing Evaluations of Summary Content from Linguistic Quality", *Proc. of the 22nd CICLing Conf.*, 2008, pp. 145–152.

[3] P. A. Rankel *et al.*, "Better Metrics to Automatically Predict the Quality of a Text Summary", *Algorithms*, vol. 5, no. 4, pp. 398–420, 2012. http://dx.doi.org/10.3390/a5040398

[4] C. Y. Lin, "Summary Evaluation Environment", 2001. http://www.isi.edu/publications/licensed-sw/see/

[5] A. Nenkova and R. Passonneau, "Evaluating Content Selection in Summarization: The Pyramid Method", *Proc. of NAACL HLT*, 2004, pp. 145–152.

[6] C. Y. Lin, "Rouge: a Package for Automatic Evaluation of Summaries", *Proc. of the ACL-04 Workshop: Text Summarization Branches Out*, 2004, pp. 74–81.

[7] E. Hovy *et al.*, "Automated Summarization Evaluation with Basic Elements", *Proc. of 4th Conference on LREC*, 2006, pp. 899–902.

[8] G. Giannakopoulos *et al.*, "Summarization System Evaluation Revisited: *N*-gram Graphs", *ACM Trans. Audio, Speech, Language Process*, vol. 5, no. 3, pp. 1– 39, 2008. http://dx.doi.org/10.1145/1410358.1410359

[9] G. Giannakopoulos and V. Karkaletsis, "AutoSummENG and MeMoG in Evaluating Guided Summaries", *Proc. of 4th TAC Conf.*, 2011.

[10] A. Louis and A. Nenkova, "Automatically Assessing Machine Summary Content without a Gold Standard", *Computational Linguistics*, vol. 39 no. 2, pp. 267–300, 2013. http://dx.doi.org/10.1162/COLI_a_00123

[11] A. Cohan and N. Goharian, "Revisiting Summarization Evaluation for Scientific Articles", *Proc. of 11th Conference on LREC*, 2016, pp. 806–813.

[12] R. Barzilay and M. Lapata, "Modeling local coherence: An entity-based approach", *Computational Linguistics*, vol. 34, no. 1, pp. 1–34, 2008. http://dx.doi.org/10.3115/1219840.1219858

[13] J. C. K. Cheung and G. Penn, "Entity-Based Local Coherence Modeling Using Topological Fields", *Proc. of 48th Ann. Meeting of the ACL*, 2010, pp. 186–195.

[14] M. d. S. Dias *et al.*, "Using Rhetorical Structure Theory and Entity Grids to Automatically Evaluate Local Coherence in Texts", *Proc. of PROPOR*, 2014, pp. 232–243. http://dx.doi.org/10.1007/978-3-319-09761-9_26

[15] R. Vadlapudi and R. Katragadda, "Quantitative Evaluation of Grammaticality of Summaries", *Proc. of CICLing*, 2010, pp. 736–747. http://dx.doi.org/10.1007/978-3-642-12116-6_62

[16] E. Pitler *et al.*, "Automatic Evaluation of Linguistic Quality in Multi-Document Summarization", *Proc. of Annual Meeting of Association for Computational Linguistics*, 2010, pp. 544–554.

[17] Z. Lin *et al.*, "Combining Coherence Models and Machine Translation Evaluation Metrics for Summarization Evaluation", *Proc. of Annual. Meeting of Association for Computational Linguistics*, 2012, pp. 1006–1014.

[18] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing", *Proceedings of the 41st Annual Meeting of Association for Computational Linguistics – vol 1*, 2003, pp. 423–430. http://dx.doi.org/10.3115/1075096.1075150

[19] K. Toutanova *et al.*, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network", *Proc. of NAACL HLT*, 2003, pp. 252–259. http://dx.doi.org/10.3115/1073445.1073478

[20] J. R. Finkel *et al.*, "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling", *Proc. of 43rd Annual Meeting of Association for Computational Linguistics*, 2005, pp. 363–370. http://dx.doi.org/10.3115/1219840.1219885

[21] H. Lee *et al.*, "Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task", *Proc. of 5th Conference on Computational Natural Language Learning: Shared Task*, 2011, pp. 28–34.

[22] A. Stolcke, "SRILM – an Extensible Language Modeling Toolkit", *Proc. of Int. Conf. on Spoken Language Processing*, vol. 2, 2002, pp. 901–904.

[23] S. Ellouze *et al.*, "An Evaluation Summary Method Based on a Combination of Content and Linguistic Metrics", *Proc. of Recent Advances in Natural Language Processing conf.*, 2013, pp. 245–251.

[24] S. Ellouze *et al.*, "Automatic Evaluation of a Summary's Linguistic Quality", *Proc. of 21st Int. Conf. on Applications of Natural Language to Information Systems*, 2016, pp. 392–400. http://dx.doi.org/10.1007/978-3-319-41754-7_39

[25] S. Kullback and R. A. Leibler, "On Information and Sufficiency", *Annual of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. http://dx.doi.org/10.1214/aoms/1177729694

[26] J. Lin, "Divergence Measures Based on the Shannon Entropy", *IEEE Trans. Inf. Theory*, vol. 37, pp. 145–151, 1991.

[27] J. J. Li and A. Nenkova, "Fast and Accurate Prediction of Sentence Specificity", *Proc. of 29th Conf. on Artificial Intelligence*, 2015, pp. 2281–2287.

[28] C. J. Huang *et al.*, "Application of Wrapper Approach and Composite Classifier to the Stock Trend Prediction", *Expert Syst Appl*, vol. 34, no. 4, pp. 2870–2878, 2008. http://dx.doi.org/10.1016/j.eswa.2007.05.035

[29] R. Kohavi, and G. H. John, "Wrappers for Feature Subset Selection", *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997. http://dx.doi.org/10.1016/S0004-3702(97)00043-X

[30] I. H. Witten *et al.*, "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, San Francisco, CA, USA, 3rd edition, 2011.

[31] L. Breiman, "Bagging Predictors", *Machine Learning*, vol. 24, pp. 123–140, 1996. http://dx.doi.org/10.1007/BF00058655

[32] L. I. Kuncheva, "Combining Pattern Classifiers: Methods and Algorithms", Second edition Wiley-Interscience, 2014. http://dx.doi.org/10.1002/0471660264

[33] D. H. Wolpert, "Stacked generalization", *Neural Networks*, vol. 5, pp. 241–259, 1992. http://dx.doi.org/10.1016/S0893-6080(05)80023-1

[34] K. Pearson, "Mathematical Contributions to the Theory of Evolution, II: Skew Variation in Homogeneous Material", *Philosophical Transactions of Royal Society London (A)*, vol. 186, pp. 343–414, 1895. http://dx.doi.org/10.1098/rspl.1894.0147

[35] H. T. Dang and K. Owczarzak, "Overview of the TAC 2008 Update Summarization Task", *Proc. of Text Analysis Conf.*, 2008, pp. 10–23.

[36] C. E. Spearman, "Correlation Calculated from Faulty Data", *British Journal of Psychology*, vol. 3, 271–295, 1910. http://dx.doi.org/10.1111/j.2044-8295.1910.tb00206.x

[37] M. G. Kendall, "A New Measure of Rank Correlation", *Biometrika*, vol. 30, 1938, pp. 81–89. https://doi.org/10.1093/biomet/30.1-2.81

[38] R. Gunning, "The Techniques of Clear Writing", (Rev. ed.), McGraw-Hill, New York, 1968.

[39] R. F. Flesch, "How to Test Readability", Harper & Brothers, New York, 1951.

[40] J. P. Kincaid *et al.*, "Derivation of New Readability Formulas for Navy Enlisted Personnel", Research Branch Report 8–75, U.S. Naval Air Station, Memphis, 1975.

[41] E. Smith, and R. Senter, "Automated Readability Index", *AMRL-TR, Aerospace Medical Research Laboratories (6570th)*, 1967, pp. 1.

[42] C. G. Atkeson *et al.*, "Locally Weighted Learning", *Artif. Intell.*, vol. 11, no. 1, pp. 11–73, 1997. http://dx.doi.org/10.1007/978-94-017-2053-3_2

Received: September 2016
Revised: May 2017
Accepted: May 2017

Contact addresses:
Samira Ellouze
University of Sfax
Faculty of Economics and Management of Sfax
Road of the Airport Km 4
3018 Sfax
Tunisia
e-mail: ellouze.samira@gmail.com

Maher Jaoua
University of Sfax
Faculty of Economics and Management of Sfax
Road of the Airport Km 4
3018 Sfax
Tunisia
e-mail: maher.jaoua@fsegs.rnu.tn

Lamia Hadrich Belguith
University of Sfax
Faculty of Economics and Management of Sfax
Road of the Airport Km 4
3018 Sfax
Tunisia
e-mail: l.belguith@fsegs.rnu.tn

Samira Ellouze is pursuing her PhD in Computer Science at University of Sfax, Tunisia. She received the Master's degree from the same University in Information Systems and New Technologies in 2010. She is a member of ANLP-RG and MIRACL Laboratory. Her research interests include natural language processing tools and especially text summary evaluation tools.

Maher Jaoua received his PhD degree from University of Tunis El Manar in 2004. He is currently an assistant professor at the Faculty of Economics and Management of Sfax at Sfax University, Tunisia. He is a member of ANLP-RG and MIRACL Laboratory. He has authored more than 20 refereed international journals and conference papers. His research interests include natural language processing, automatic summarization, automatic text summary evaluation, Information Retrieval System, author profiling, automatic comprehension of Tunisian dialect, etc.

Lamia Hadrich Belguith is a Professor of Computer Science at the Faculty of Economics and Management of Sfax (FSEGS) - University of Sfax (Tunisia). She is Head of Arabic Natural Language Processing Research Group (ANLP-RG) of Multimedia, Information Systems and Advanced Computing Laboratory (MIRACL). She has authored more than 200 refereed international journals and conference
papers. Her research interests are natural language processing, artificial intelligence, machine translation, text mining, Arabic text analysis, automatic abstracting, question-answering systems, Arabic human-human discussion processing, automatic comprehension of Tunisian dialect, etc.