

# Understanding Summaries: Modelling Evaluation in Extractive Summarisation Techniques

Victor Margallo <sup>a</sup>

*PublicSonar, Zuid Hollandlaan 7, The Hague, The Netherlands*

**Keywords:** NLP, Extractive Summarisation, Evaluation, Summary Quality Modelling.

**Abstract:** In the task of providing extracted summaries, the assessment of quality evaluation has been traditionally tackled with n-gram, word sequences, and word pairs overlapping metrics with human annotated summaries for theoretical benchmarking. This approach does not provide an end solution for extractive summarising algorithms as output summaries are not evaluated for new texts. Our solution proposes the expansion of a graph extraction method together with an understanding layer before delivering the final summary. With this technique we strive to achieve a categorisation of acceptable output summaries. Our understanding layer judges correct summaries with 91% accuracy and is in line with experts' labelling providing a strong inter-rater reliability (0.73 Kappa statistic).

## 1 INTRODUCTION

One of the multiple applications of Natural Language Processing is summarisation. The goal of such a task is to extract or generate a shorter version of the original text. There are two approaches for summarisation; the first one is extractive and the second one is generative.


Extractive summarisation makes use of literal sentences or keywords from the original text ranking them with an importance metric. On the other hand, generative models rely predominantly on Deep Learning techniques that decode the original text into a shorter version, mimicking human summaries shown in the training phase. Even though the latter strikes to yield a more human-like synthesis, extractive summarisation is unsupervised and the only technique applicable when dealing with no training datasets. Due to the lack of availability in summarisation datasets for minor languages, extractive methods are still of great relevance and the main driver for this research in the Dutch language.

Methods to address evaluation of summarising output are limited to overlapping metrics based on n-gram, word sequences, and word pairs ratios. These metrics described in methods as ROUGE (Lin, 2004) present drawbacks such as the ambiguity of the ground truth depending on the annotators and the lack

of evaluation on newly summarised text. Even though evaluation on new text is not a concerning issue in theoretical research of the model—as it is evaluated against a predefined ground truth—it becomes critical in a real world usage of the algorithm since it is to be executed on new text with no manual annotation for its quality evaluation. More recent research, as in Wu et al. (2020), includes advanced text embedding comparisons in order to provide a score, also in new summaries. Yet, it is not defining a line regarding quality acceptability. Hence, a gap between the research usage and the practical usage of summarising algorithms exists. As a result of a missing quality check, extractive algorithms can deliver unsuitable summaries for human end-readers, drawing the purpose of a summarising technique—this is, shortening input text into an output text that is concise, readable and understandable—away.

We propose an integration of a graph model together with a layer replicating the understanding criteria for a readable and correct summary. This pipeline provides the summary, firstly created by the graph model, as long as the understanding layer confirms it is readable. We achieve 91% accuracy in this task. Plus, the results show the algorithm to be in consistent agreement with the variability of the human assessment concerning summary output quality, measured with an inter-rater reliability Kappa value of 0.73.

The paper follows with Section 2 introducing related work in the field, focusing on the nature of ex-

<sup>a</sup>  <https://orcid.org/0000-0002-5765-6671>

tractive summarising algorithms and the main model evaluation method ROUGE; Section 3 describes the data used in this research together with the criteria established for the definition of a good quality summary, the algorithmic architecture for the solution—from the graph model to the developed understanding layer and its optimization—and the construction of input features from source text and output text that feed the understanding layer; finally, Section 4 and Section 5 show the results of the research and end it with a conclusion and future work.

## 2 RELATED WORK

Extractive summarisation sets its foundations on graph-based ranking algorithms such as PageRank (Brin and Page, 1998) and HITS (Kleinberg, 1999). These algorithms were successfully applied to citation analysis, link-network of webpages and social networks. The translation of these approaches to natural language tasks rendered algorithms as TextRank (Mihalcea and Tarau, 2004) and progressive refinements as in Barrios et al. (2016).

Graph-based methods utilize the holistic knowledge of the text of interest in order to make local ranking of sentences or words. With a structured ranking of sentences the algorithm selects a reduced version of the original text containing a presupposed high relevancy.

The TextRank method uses the importance of vertex connections to create a final score. To formalize, given that  $G = (N, E)$  is a directed graph with nodes  $N$  and edges  $E$ . Edges are connections between nodes and thus, a  $N \times N$  subset. For a  $N_i$  node, TextRank sets  $In(N_i)$  as predecessor nodes pointing at the current  $N_i$  node and  $Out(N_i)$  as the nodes it points out to. The score of  $N_i$  is defined as:

$$S(N_i) = (1 - d) + d * \sum_{j \in In(N_i)} \frac{1}{|Out(N_j)|} S(N_j) \quad (1)$$

Where  $d$  represents a damping factor to account for the *random surfer model* probability Brin and Page (1998). However, sentence units in natural text relate to each other with similarity scores. TextRank reshapes (1) so that a  $W$  factor captures it:

$$WS(N_i) = (1 - d) + d * \sum_{W_j \in In(N_i)} \frac{w_{ji}}{\sum_{N_k \in Out(N_j)} w_{jk}} WS(N_j) \quad (2)$$

The score of sentences and therefore the weights of the edges between sentences is given by the overlap described as:

$$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)} \quad (3)$$

Where  $S_i$  and  $S_j$  are composed of tokens  $S_i = w_1^i, w_2^i, \dots, w_{N_i}^i$  that represent processed words. The resulting most significant sentences—those with highest ranks—form the final summary representation.

In the succession of research in this field, most effort was focused on improving a key component of the algorithm—the similarity score. Barrios et al. (2016) show how the use of cosine distance with TF-IDF and BM25 improve the quality of the results. Yet, one of the drawbacks of TextRank algorithms is their disability to address the suitability of incoming natural text. Consequently, it always produces an output regardless of what input text is given. This results into poor quality and low readability summaries for some texts as it is illustrated in Figure 1.

Concerning the evaluation of the output, Mihalcea and Tarau (2004) appraise their method using the ROUGE technique. Lin (2004) presents ROUGE as a solution to the expensive and difficult process of human judgement on evaluating the different factors of a resulting summary. ROUGE employs a set of reference summaries extracted by humans. The algorithmic solutions are then compared to the human ones by means of co-occurrence statistics. The problem with this approach is that we assume the reference summary as the ground truth. However, there may be several combinations which provide a good quality summary that still do not match the human's approach (Mani, 2001). Therefore, even though ROUGE-like metrics deliver a reliable benchmarking for fair comparison, with this paper we expand the evaluation method by answering the question *Can we detect a human acceptable summary output?*

We observe that, when applying these TextRank-based algorithms, it is uncertain whether results are summaries of good or bad quality. Given an original text being noisy and not formally structured, the resulted summary will be of the expected same low quality. Likewise, if the original text does not contain clear sentences that help summarize the content, the output will not make sense to the reader. We can think of an interview article formatted in dialogues. Extractive algorithms are limited to selecting sentences and, in scenarios such as a dialogue/interview, there are no good candidates to form a complete summary. Another example is financial news which touch upon many topics. Last example refers to sports articles; we may face a long article describing all the matches of the day. The output will not be satisfactory on summarising well all the content information. Some examples of the above mentioned situations are listed in Figure 1.

Thus, in this paper, we aim at the definition of quality and readability for summaries delivered by

		Dutch	English
Financial	Title	Spanningen doen beleggers schuilen in obligaties	Tensions hide investors in bonds
	Summary	De aanslagen deden donderdag de olieprijs met 2,2 procent stijgen.	The attacks on Thursday caused the oil price to rise by 2.2 percent.
Sports	Title	Rondom: Ajax neemt afscheid van Sinkgraven	All around: Ajax says goodbye to Sinkgraven
	Summary	De Ajax-verdediger trouwde afgelopen weekend met zijn vriendin Candy Rae Fleur en nu heeft hij wederom goed nieuws te melden: Blind wordt namelijk vader. Voor nu past hij bij het Braziliaanse nationale elftal in ieder geval netjes op Ajax-aanvaller David Neres.	The Ajax defender married his girlfriend Candy Rae Fleur last weekend and now he has good news again: Blind will become a father. For now he fits nicely with the Brazilian national team on Ajax attacker David Neres.
Interview	Title	'Verrassing, je bent opa geworden!'	"Surprise, you became a grandpa!"
	Summary	Zoals bij Angelique van de Ven-Herbergs (33) uit Maarheeze, die dolblij is dat ze op deze dag haar vader nog heeft. Dertien jaar geleden kreeg hij een hartinfarct. En daarom is Van de Ven ieder jaar weer dankbaar. Afgelopen Vaderdag was de meest bijzondere voor haar vader Wim. En Van de Ven zou eigenlijk op die Vaderdag met haar familie, inclusief vader, gaan lunchen. Voor Van de Ven is een Christoffel-sleutelhanger met als tekst 'Kom Veilig Thuis', die ze even daarvoor van haar vader had gekregen, haar dan ook heel dierbaar. "Mijn vader is zo 'n lieve gezellige man, waarbij je vroeger als kind op schoot kroop om samen naar Duck Tales te kijken op tv." Ook voor Jolanda Walet uit Soest en haar 92-jarige vader Arend is Vaderdag altijd heel bijzonder. "Mijn vader zal de Vaderdag van vorig jaar niet gauw vergeten", vertelt Walet. "De avond voor Vaderdag kreeg ik opeens het idee om hem te gaan verrassen. Ik vertelde één van de medewerkers mijn verhaal en ze leefde zo mee dat ik direct mocht doorlopen." Eenmaal aangekomen in Benidorm bleek haar vader echter niet thuis te zijn. Genoeg reden voor haar om nog dankbaar terug te blikken op de afgelopen jaren met haar vader.	Such as Angelique van de Ven-Herbergs (33) from Maarheeze, who is overjoyed that she still has her father to this day. Thirteen years ago he had a heart attack. And that is why Van de Ven is grateful every year. Last Father's Day was the most special for her father Wim. And Van de Ven was actually going to have lunch with her family, including father, on that Father's Day. For Van de Ven, a Christoffel key ring with the text 'Come Safe Home', which she had just received from her father, is very dear to her. "My father is such a sweet and sociable man, where you used to crawl on your lap as a child to watch Duck Tales on TV together." Also for Jolanda Walet from Soest and her 92-year-old father Arend, Father's Day is always very special. "My father will not soon forget last year's Father's Day," says Walet. "The evening before Father's Day, I suddenly had the idea to surprise him. I told one of the employees my story and she sympathized so much that I was allowed to continue immediately." Once arrived in Benidorm, however, her father turned out not to be home. Enough reason for her to look back with gratitude on the past years with her father.

Figure 1: Examples of bad quality summaries.

TextRank methods and hereby propose an understanding layer to avoid poor output and solve its non-discriminant nature.

### 3 METHOD

Hereinafter, we proceed to describe our modelling methodology. The following subsections will consist on the annotated data utilized to model decision making on the summary output; the algorithmic architecture of our solution; and the construction of input features from both incoming and outgoing text.

#### 3.1 Data

Our dataset is composed of 500 Dutch annotated documents forming our ground truth. The articles contained in our sample have been randomly selected in a pool of different news sources with a diverse range of topics.

A group of experts annotated the sample dataset following guidelines in pursuit of a harmonious an-

notation result. The annotation guidelines converge the assessment of the summary quality in the pillars of *readability*, *information* or *content*, and *coherence*.

*Readability* refers to the lack of textual barriers or noise—for instance, misplacement or presence of unsought characters or wrong parsing. *Information* or *content* introduce the rules in which a summary is considered correct when it delivers a general understanding of the main topic of the article. Finally, *coherence* covers the complete sense of the text. Hence, it implies the holistic correctness of the summary—this includes the absence of unrelated or incongruous sentences in the summary.

Additionally, we implement an anonymous feedback mechanism so that reviewers can vote on the summary independently. A majority vote decides the final label for the summary. Herewith we hope to alleviate annotators' bias in the assessment of the summary. One potential bias is, for instance, the familiarity of the annotator with the topic of the text and thus, his better understanding of a small context compared to an uninformed annotator. Despite these efforts, it is safe to assume discrepancies in human judgement

towards the quality of the summary and therefore, we include the Kappa statistic (Cohen, 1960; McHugh, 2012) to consider the inter-rater degree of agreement. The Kappa statistic calculates the agreement between two sets of annotations or evaluations considering the possibility of random agreement.

### 3.2 Algorithmic Architecture

Our graph extraction pipeline is based on the algorithm of Barrios et al. (2016) and works on the sentence level of the text. We optimize the preprocessing pipeline for the Dutch language, specifically, the stopwords list curation, the sentence splitter and the stemmer. The graph extraction model ranks processed sentences on their centrality values:

$$\text{centrality}(S_1, S_2) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, S_1) \cdot 2.2}{f(q_i, S_1) + 1.2 \cdot (1 - 0.75 + 0.75 \cdot \frac{|S_1|}{\text{avgsl}})} \quad (4)$$

Equation 4 shows the calculation of centrality for  $S_1$  and  $S_2$  which represent a pair of sentences in the text. The variable  $q_i$  are the terms of  $S_2$ .  $\text{IDF}(q_i)$  is the function computing the Inverse Document Frequency (Jones, 1972) of the term  $q_i$ . In order to avoid non-valid IDF values for terms out of the vocabulary, the function has a floor given by  $0.25 \cdot \text{avgIDF}$  for those cases. The function  $f(q_i, S_1)$  is the term frequency of  $q_i$  in  $S_1$ . The length of  $S_1$  in words is  $|S_1|$  and  $\text{avgsl}$  is the average sentence length in the original document.

The most central sentences, ordered by appearance on the original article, compose the summary. The number of sentences will be parametrized as a preset percentage of the original length. In our research, the ratio of summary sentences to original sentences is set at 0.15.

Our understanding layer forms the second part of the algorithm. This layer mimics a quality evaluator in order to exclusively pass through readable and comprehensible text. Three different models define the architecture of this layer. An ensemble of Random Forest, Support Vector Machine and a Neural Network. The reason to use an ensemble of models is to exploit the strengths of each individual component. The strategy of bringing different classifiers together provides an improvement on the generalization performance (Güneş et al., 2017). Plus, the combinations of outputs reduces the probability of choosing a poor classifier and the average error rate. Namely, following Wolpert (1992), assuming a constant error rate  $\epsilon$  and the independence of classifiers, the error rate of the ensemble benefits from the diversification effect shown by:

$$e_{ensemble} = \sum_n \binom{N}{n} \epsilon^n (1 - \epsilon)^{N-n} \quad (5)$$

Where  $N$  is the number of classifiers in the ensemble model and  $n$  is the majority voting number. The term  $\epsilon$  is the error rate. One of the requirements in the selection of the components is the diversity between them. This diversity in the classifiers will reinforce the independence assumption of the error rate made beforehand by a lower correlation in the predictions.

Random Forests are excellent stand-alone algorithms as they are composed by an ensemble of decision trees. This method bootstraps random samples and selects a random number of features. As a result, Random Forest becomes a robust classifier to outliers and noise, with a good generalization to new data and highly parallelizable (Breiman, 2001).

Support Vector Machines, on the other hand, are suited methods for binary classification that work empirically better on sparse data such as text classification problems (Hearst et al., 1998). Its nature resides on the distance of support vectors in order to draw a decision boundary.

Finally, Neural Networks are connected layers that tend to outperform Machine Learning models as the training dataset increases. Additionally, the inclusion of a sequential model allows to capture the text linearity that Bag of Words techniques in ML models fail to address (Schuster and Paliwal, 1997; Zhou et al., 2016).

In this paper we design a weighted voting ensemble model architecture. The three models previously mentioned are combined with different input features explained in the next section. The ensemble model is exposed to a Monte Carlo simulation selecting training and validation data. The process is performed thousand times to assure the covering of most training/validation splits and help minimize the bias of the estimates following Zhang (1993). The probability results of the ensemble model are eventually optimized by:

$$\max(AUC) = \max \left( \int_{x=0}^1 \text{TPR}_i(\text{FPR}_i^{-1}(x)) dx \right) \quad (6)$$

Where TPR is the True Positive Rate and FPR is the False Positive Rate for every  $i$  weight distribution of the ensemble models. Formally, we define both rates as:

$$\text{TPR}_i(T) = \int_T^\infty f_1^i(x) dx, \quad \text{FPR}_i(T) = \int_T^\infty f_0^i(x) dx \quad (7)$$

Where  $T$  is the probability threshold to classify the prediction  $X$  and under the conditions defined by:

$$f_1^i(x) = \{X_j | X_j > T\}, \quad f_0^i(x) = \{X_j | X_j < T\} \quad (8)$$

The probability density function is  $f_1(x)$  for positive predictions and  $f_0(x)$  for the negative ones. We define  $X_j$  as:

$$X_j = (w_{SVM}P_{SVM}^j + w_{RF}P_{RF}^j + w_{NN}P_{NN}^j) \quad (9)$$

Where  $j$  is the different data points in the validation set and  $(w_{SVM}, w_{RF}, w_{NN})$  belong to a set  $W$  containing all combinatory possibilities for the weight distribution of the ensemble model.

The *AUC* or *Area Under the Curve* shows the performance by comparing the TPR versus FPR trade-off. In an intuitive way, the result of *AUC* is the probability of a random positive data point ranking higher than a negative one. Section 4 will visualize the optimal *AUC* in the *Receiver Operating Characteristic curve* or *ROC curve* (Bradley, 1997). *ROC* is the graphical representation of the *AUC* trade-off, plotting the TPR against the FPR levels.

### 3.3 Input Features

Our input features are built in order to provide the differentiable aspects of the text so that the ensemble model can determine the correct class.

There are three pillars—the formatting, the semantics and the syntax. The formatting of the text includes the length of the text measured on the number sentences, the average sentence length in characters, average number of words per sentence, and number of dialogue dashes, colons, quotes, question and exclamation marks. These features are specially determinant on texts containing complex formats on which the summarisation algorithm fails and, consequently, translates to a non-readable summary. The semantic layer helps define how likely a text is to be correct or incorrect given its thematic. For instance, sports and interview articles are prone to produce unsatisfactory summaries. The syntax layer considers what an acceptable concatenation of syntactic units is. This helps detect when sentences are wrongly extracted—we may think of two sentences put together due to a mistake in the sentence splitter in the pre-processing. As a result of its sequential nature, this input is only available to the LSTM layers of the Neural Network model.

Following Figure 2, we store the extracted formatting features in a Coordinate list (COO) with (row, column, value) structure, this will help coordinate the stacking with the semantic layer extracted through TF-IDF (Jones, 1972) stored in Compressed sparse row (CSR). Furthermore, we parameterize the weight of each array before concatenating. This is defined by  $\alpha$  in the formatting array and  $\beta$  in the semantic array. In the background  $\alpha$  and  $\beta$  are subdivided into two more parameters  $(\epsilon_v, \theta_v)$  for  $v = \alpha, \beta$  that provide the weight distribution to arrays from the original text and arrays from the summarized text. This sparse array is fed to the Random Forest and to the Support

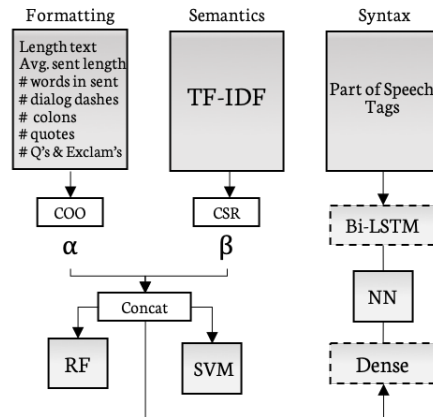


Figure 2: Input features.

Vector Machine as well as to the dense layer of the Neural Network. Syntax is captured by a fixed length sequence containing the Part of Speech tags from only the summary text. This feature is read left-to-right and right-to-left by a Bidirectional LSTM (Bi-LSTM) (Schuster and Paliwal, 1997).

## 4 RESULTS

Input weights  $\alpha$  and  $\beta$  are fine-tuned on cross-validation with grid search during warm-up simulations. The ensemble model weight distribution is assigned to the predicted probabilities from each of the models and optimized based on equation (6). Figure 3 shows the result of such optimization, where the best performing weight distribution is marked with red and visualises the True Positive Rate trade-off with the False Positive Rate of the ensemble for the different threshold  $T$  as explained in (7). The blue channel in Figure 3 represents all other weight combinations arisen from the simulations. Lastly, the grey area is the baseline for a random guess.

Table 1 exhibits the evaluation metrics for the optimized understanding layer at  $T = 0.5$ . Our ensemble model obtains a conclusive 90.84% accuracy on our validation rounds. The ensemble model seems to weaken in recalling *bad quality summaries*, such lower recall determines consequently the impact on the precision metric of *good quality summaries* due to its binary outcome. This flaw in *bad quality summaries* recall can be explained by the smaller amount of such examples in the training phase. Overall, the results in Table 1 show that it is possible to transmit quality checks through an engineered formatting layer together with the syntax and the semantics. These findings satisfy our objective to create an algorithmic solution that could substitute or emulate expensive manual evaluation.

Table 1: Understanding layer: Evaluation metrics.

Label	Precision (Std.) %	Recall (Std.) %	F1 (Std.) %	Samples support	Support pct. %
Bad quality	98.65 (2.70)	64.63 (7.60)	78.09 (5.80)	3158	25.26
Good quality	89.29 (2.72)	99.70 (0.64)	94.21 (1.56)	9342	74.73
Accuracy (Std.) %	90.84 (2.33)				

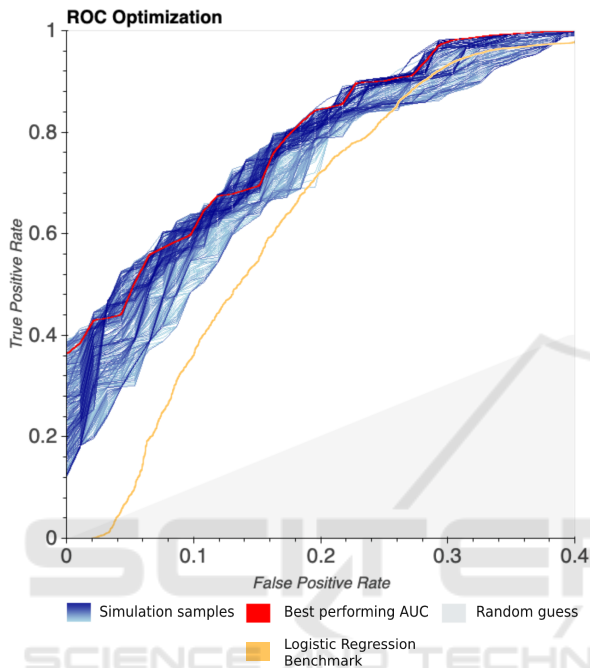


Figure 3: Optimization for model distribution.

Nevertheless, and despite the usefulness of the previous metrics for reference and model fine-tuning, we face an evaluation with different outcomes based on the specific individuals assessing the quality of the final summary. This lack of agreement produces potential mismatches on the same summary outcome. Thus, it is crucial to quantify the consensus between the labelling experts and the algorithmic solution to judge the performance of our understanding layer in perspective. Therefore, we proceed to evaluate the variability in the data labeled by the experts.

We introduce Cohen's Kappa statistic ( $\kappa$ ) (Cohen, 1960) to measure the degree of agreement or disagreement happening by chance. It ranges from -1 to 1 where 0 is random chance and 1 represents perfect agreement.

In our experiment, we merge our experts' labels and we include the algorithm as a second rater in order to model the raters' agreement probabilities. In Table 2 the results for our model are listed. Table 2 are the agreement probabilities calculated from our dataset and from the model's results. The latter proba-

Table 2: Results of Kappa statistic through agreement probabilities.

Agreement probabilities	$P_o$	$P_{max}$	$P_{Good}$	$P_{Bad}$	$P_e$
Values (%)	90.84	91.29	62.37	4.18	66.55

(a) Agreement probabilities from our testing dataset.

Kappa statistics	$\kappa$	$\kappa_{max}$	$SE_{\kappa}$	CI (95%)
Values	<b>0.7262</b>	0.7395	0.0077	[0.7185,0.7339]

(b) Kappa statistics calculated from the agreement probabilities.

Table 3: Kappa agreement levels.

Std. range	<0	<0.2	<0.4	<0.6	<0.8	<1
Norm. range	<0	<0.15	<0.3	<0.44	<0.59	< <b>0.74</b>
Agreement magnitude	None	Slight	Fair	Moderate	Substantial	<b>Almost perfect</b>

Note: Magnitude assessment following Landis and Koch (1977).

bilities are the input for the calculus of the Kappa values in Table 2b. In Table 3 we provide the standard range for equally distributed categories and the normalised range for our specific sample. The standard range is used as a Kappa benchmark established by Landis and Koch (1977) in order to express the agreement magnitude for different Kappa values. The normalised range takes into account the unbalanced distribution of our classes and scales the standard range based on the maximum Kappa value. Our understanding layer achieves a 0.7262 Kappa statistic of a maximum of 0.7395. By considering chance agreement in an ambiguous qualitative task such as summary evaluation, the finding of a strong Kappa (0.7262) shows a behaviour almost completely in line with the expected behaviour from a human data labeler.

## 5 CONCLUSION AND FUTURE WORK

In the lack of quality assessment of extractive summaries, we present an understanding layer in order to determine the readability of the outcome. We have shown how we may translate human assessment of summaries output into a modelling process that suc-

successfully performs the task with 91% accuracy. Furthermore, our Kappa statistic (0.73) reinforces a consistent agreement between algorithm's and expert's summary labelling. This solution links the evaluation of co-reference solutions for benchmarking (Lin, 2004) to an applicable solution for real summary understanding.

Future research will focus on the readability of the models. This refers to whether we may have a snapshot from the ensemble model of the main features delivering the final decision. In other words, we want to be knowledgeable of the relevance of the formatting, semantic and syntactic layers on the result. This would elucidate the understanding of the potential for new applications regarding the machine-human correlation on decision making in the task of summary evaluation.

Lastly, the scope of this study is limited to the Dutch language. It is to be expected a similar performance in close relative languages such as German and English, yet challenges increase in more distant types. Therefore, modelling techniques should be fine-tuned and adapted to each specific language to validate the results previously presented.

## REFERENCES

- Barrios, F., Lopez, F., Argerich, L., and Wachenchauser, R. (2016). Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.
- Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Güneş, F., Wolfinger, R., and Tan, P.-Y. (2017). Stacked ensemble models for improved prediction accuracy. In *Proc. Static Anal. Symp.*, pages 1–19.
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.
- Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632.
- Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text summarization branches out*, page 74–81.
- Mani, I. (2001). *Automatic summarization*. J. Benjamins Pub. Co.
- McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, page 276–282.
- Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, page 404–411.
- Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.
- Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.
- Wu, H., Ma, T., Wu, L., Manyumwa, T., and Ji, S. (2020). Unsupervised reference-free summary quality evaluation via contrastive learning. *arXiv preprint arXiv:2010.01781*.
- Zhang, P. (1993). Model selection via multifold cross validation. *The Annals of Statistics*, 21(1):299–313.
- Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., and Xu, B. (2016). Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.

## APPENDIX

In Table 2 we define  $p_o$  as the raters' accuracy. The maximum agreement probability is  $P_{\max} = \sum_{i=1}^k \min(P_{i+}, P_{+i})$  where  $P_{i+}$  and  $P_{+i}$  are the row and column probabilities from the original raters' matrix.  $p_{Good}$  and  $p_{Bad}$  are the probability of random agreement for the different summary categories.  $p_e$  is the random probability for both categories together, thus,  $p_e = p_{Good} + p_{Bad}$ . Kappa value is defined as  $\kappa = \frac{p_o - p_e}{1 - p_e}$ . The maximum value for the unequal distribution of the sample is  $\kappa_{\max}$  and is calculated by  $\kappa_{\max} = \frac{P_{\max} - p_e}{1 - p_e}$ . The standard error and confidence interval are calculated by  $SE_{\kappa} = \sqrt{\frac{p_o(1-p_o)}{N(1-p_e)^2}}$  and  $CI : \kappa \pm Z_{1-\alpha/2} SE_{\kappa}$  respectively.