Beyond "Using Their Own Words": Abstractivity Characterization in Summarization

Anonymous ACL submission

Abstract

In this work, we present an extension of the definition of abstractivity within the scope of the automatic generation of summaries. We propose to join extractivity and abstractivity in a single dimension, where extractivity would be on one side of the dimension and complete abstractivity on the opposite one, but in between, there would be levels of abstractivity. A dataset manually annotated to characterize the level of abstractivity of the summaries and to measure the presence of a set of actions applied to compose the summaries has been built. Using this dataset, a study of the sample distribution in terms of abstractivity, annotator agreement, and correlation between annotations regarding the set of actions is presented. An experimental work with a double objective is carried out; on the one hand, we want to validate our perception that extractivity and complete abstractivity are extreme points of a single dimension with multiple abstractivity levels, and on the other hand, we want to verify if there is an overall correlation between the frequency of the actions used for creating the summary and the level of abstractivity. The results confirm both objectives.

1 Introduction

Summarizing is the process of condensing the most relevant information from a document into a single, shorter document, the summary. Initially, the essential information in the article has to be identified. There are two strategies to generate the summary from the selected information. In an extractive approach, the sentences with the selected information are copied directly to the summary. In an abstractive approach, the generated summaries also contain the essential information, but it is "expressed, usually, in the words of the author of the summary" (Nenkova and McKeown, 2011).

Although the first approaches to the problem were extractive, after the emergence of the Transformer architecture (Vaswani et al., 2017) and its capabilities, most of the published works have addressed the generation of summaries under abstractive approaches. However, to the best of our knowledge, the characterization of abstractivity within summaries has not been sufficiently studied (Bommasani and Cardie, 2020; Grusky et al., 2018; Kryściński et al., 2018; Jing, 2002). A more detailed and extended characterization of the abstractivity in summaries would help to better understand how abstractive models generate their summaries.

Generally, works related to the evaluation of the level of abstractivity of the generated summaries focus on measuring the appearance of new words in the summaries compared to the summarized documents (Wu et al., 2021; Chen et al., 2021; Fu et al., 2021; Manakul and Gales, 2021; Dou et al., 2021; Zou et al., 2020; Zheng et al., 2020). This strategy conforms to Nenkova and McKeown's definition of abstractive summaries. However, it is not the only way to produce a summary in "the author's words". It is possible to make a summary in which very few new words or expressions are introduced compared to the original document, and yet the main ideas are expressed in a different way (Ahuir et al., 2021). For example, a summary can be written based mainly on the reordering of some segments extracted from the document, with the introduction of very few new elements.

In 2002, Jing conducted a study on the actions that abstraction professionals used to create their abstractive summaries (Jing, 2002). Specifically, he identified the following six actions: sentence reduction, sentence combination, syntactic transformation, lexical paraphrase, generalization/specification, and reordering. Based on the hypothesis that writing an abstractive summary is based on using this set of actions, we can characterize the abstractivity of a text by measuring the presence of each of the six actions.

In this work, we propose an extension of the definition of abstractivity in the automatic summarization area. Although in the literature, the extractive and abstractive approaches have been treated as mutually exclusive (Sun et al., 2024; Varab and Xu, 2023; Liu and Lapata, 2019), we join extractivity and abstractivity in a single dimension, what we call the level of abstractivity. The extractivity would be on one side of the dimension, and the complete abstractivity on the opposite one, but in between, there would be levels of abstractivity. Additionally, we want to characterize the level of abstractivity of a summary and measure the presence of each of the actions identified by Jing.

The main contributions of this work are:

(i) A dataset has been built that contains document-summary pairs manually annotated in terms of a set of actions (including the Jing's actions) using a Likert scale: the Characterization of the Level of Abstractivity in Summarization (CLAsum) dataset. It is publicly available at `https://huggingface.co/datasets/??`.

(ii) Some analyses have been carried out on the CLAsum dataset: sample distribution in terms of abstractivity, annotator agreement, and correlation between annotations in terms of the set of actions.

(iii) To check if there is an overall correlation between the frequency of the actions used for creating the summary and the level of abstractivity, two tasks have been defined: Abstractivity Inducting Features extraction, and Abstractivity Level prediction. Both tasks have been addressed as both classification and regression problems.

(iv) Using the CLAsum dataset, a set of machine learning models have been trained to predict both, the Abstractivity Inducting Features and the Abstractivity Level in summaries.

(v) Using these models, some experimentation is carried out to test how beneficial the inclusion of the Abstractivity Inducting Features information is in the Abstractivity Level prediction.

2 The CLAsum dataset

2.1 Sample Gathering

With the aim of building an appropriate dataset, we selected the test partitions of two well-known datasets in the summarization area: CNN/DailyMail (See et al., 2017), and XSum (Narayan et al., 2018). Since we wanted diversity regarding the abstractivity, we distributed the samples of both test sets into 5 clusters per source using the KMeans algorithm and selecting some features related to abstractivity. Specifically, we used the following abstractivity indicators: Coverage and Density (Grusky et al., 2018), Content Reordering (Ahuir et al., 2021), Abstractivity (p=[2,3]) (Bommasani and Cardie, 2020), and Novel [2,3,4]-grams (Kryściński et al., 2018). Therefore, an 8-component features vector was used to characterize a sample.

From those clusters, we extracted 20 samples per cluster and source. The final set comprised 100 samples from CNN/DailyMail and another 100 from XSum. To ensure the labeling process, some restrictions were required: (1) the document should contain a maximum of 500 words, (2) the summary should contain a minimum of 38 words, (3) the proportion of words document/summary should be at least 2:1.

2.2 Labeling Guideline

Since our main objective was to evaluate how the information in the document was modified (removing content, merging sentences, etc.), it is necessary to ensure that a "summary" is really a summary, that is, much of its information comes from the document, although it can provide complementary information. To detect those supposed summaries that are not really summaries, we included two previous questions: question A about the relevance of the information included in the summary with respect to the document and question B about the amount of new information added by the summary. The abstractivity-related questions were 8, from C to J. One question about the perception of abstractivity (question C) and 7 questions for the actions identified by Jing (Jing, 2002) (from D to J); Generalization (question H) and Specification actions (question I) were split to gain information. The complete guideline can be found in Appendix A.

We designed the guideline with a Likert scale. The number of options would vary from question to question since some aspects required more granularity than others. For each question, we added options until we felt that the possible answers collected enough variability and the annotators would not be forced to choose one option as a fallback. The number of options are the following ones: (A) Relevance of the information in the summary (5 options), (B) Amount of novel information within the summary (3 options), (C) Perception of the level of abstractivity (5), (D) Content exclusion (4), (E) Sentence information melting (3), (F) Syntax alteration (3), (G) Synonym usage (3), (H) Generalization usage (4), (I) Specification usage (4), and (J) Content Reordering (3).

Additionally, we included the answer 0 ("Does not apply; it is not a summary.") for questions C to J (abstractivity-related questions). In that way, the annotators would not be forced to answer the abstractivity-related questions if they do not consider the evaluated text a valid summary.

2.3 Labeling Process

The labeling process was conducted by people from our research group, a total of 13 people with a high degree level of studies in Computer Science (9 University professors, 4 PhD students, and 1 Master's degree student). Additionally, 4 Computer Science degree students collaborated with the labeling process. Thus, 17 volunteers with good English level (but not native speakers) contributed to accomplishing the annotation process.

Since we wanted to build a annotated dataset with more than one set of labels per document-summary pair, we established to obtain 3 different sets of labels per pair, acquiring a total of 600 samples (pair+labels). Also, we pursued to capture the variety of perceptions from groups of people, therefore, we distributed the samples to the annotators, avoiding the coincidence 3-annotators group between document-summary pairs as much as possible.

We provided the annotators with the ANONYM labeling tool[1] (Appendix B) and the guideline. To avoid any bias, no labeling examples or instructions were provided. We only encouraged annotators to agree on whether a document-summary contained an actual summary.

2.4 Sample Distribution

Table 1, shows the distribution of pairs that contain an actual summary and which ones do not.

Summary	Not Summary
175	25

Table 1: Distribution of document-summary pairs that contain a summary and which do not contain an actual summary (*not-summary*).

[1] https://github.com/anonym/url

We observe that 12.5% of pairs do not contain an actual summary since they do not contain at least some information extracted from the summarized document. All the *not-summaries* pairs came from the XSum dataset. Since we were studying the abstractivity in summaries, we excluded these 25 pairs from the rest of the study.

Regarding the perception of abstractivity (question C), Fig. 1 shows the distribution per source (where 1 represents the extractivity summarization style and 5 the highest perception of abstractivity).

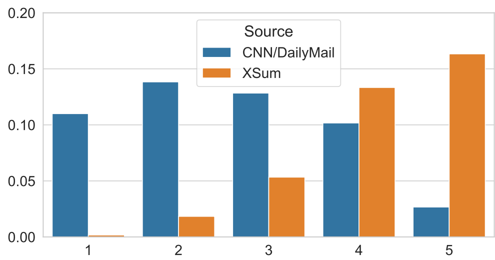

Figure 1: Distribution of answers for question C, regarding the perception of the abstractivity level in the summary.

In Fig. 1, it can be noticed that although the process of selection was the same, the perception of the level of abstractivity from source to source is different. Annotators perceived higher levels of abstractivity in the XSum dataset than in the CNN/DailyMail dataset, which presents more diversity regarding the level of abstractivity.

2.5 Annotator Agreement Analysis

The labeling process addresses a complex and subjective task. A total agreement between annotators cannot be expected, then, it would not be advisable to study the agreement in terms of exact matches (binary distance). Therefore, we used the *Relative* distance between two labels.

Eq. (1) shows the definition of this distance.

$$\text{R-Dist}_Q(l_1, l_2) = \frac{|l_1 - l_2|}{M_Q - 1} \qquad (1)$$

Given two labels (l_1, l_2) for question Q, *Relative* distance returns the percentage of the absolute distance that separates l_1 from l_2, in relation to the range between the minimum value (1) and the max value that can acquire this question (M_Q).

Table 2 shows the average agreement among annotators for each question. We used Cohen's (Cohen, 1960) and Fleis' (Fleiss, 1971) Kappa for the measurement, with the *Relative* distance (Eq. (1)) as distance function between observations.

Question	Cohen's Kappa	Fleis' Kappa
A	0.94±0.15	0.75±0.21
B	1.00±0.00	0.87±0.22
C	0.92±0.18	0.71±0.19
D	0.92±0.19	0.67±0.23
E	0.96±0.16	0.64±0.34
F	0.90±0.24	0.52±0.30
G	0.90±0.23	0.61±0.28
H	0.86±0.24	0.60±0.22
I	0.86±0.24	0.59±0.22
J	0.89±0.25	0.46±0.32

Table 2: Agreement scores per Question with the *Relative* distance. *Cohen's Kappa* is the pair-wise average score among the three annotators.

It can be observed that the average agreement with Cohen's Kappa is almost perfect. However, when Fleis' Kappa is considered, the agreement strength is reduced to substantial on most of the questions (except B), and moderate for questions F, I, and J. It can be deduced that the annotators' answers do not differ that much for a given question; however, there are slight degree deviations among the three annotations at once (the answers are not unanimous).

We extracted the distances between annotators and questions for each document-summary pair's question to analyze the deviations between annotators. The integer distance was measured between two answers; the distance was computed by counting the number of answers that separated one label from the other. Fig. 2 shows the distribution of integer absolute distance.

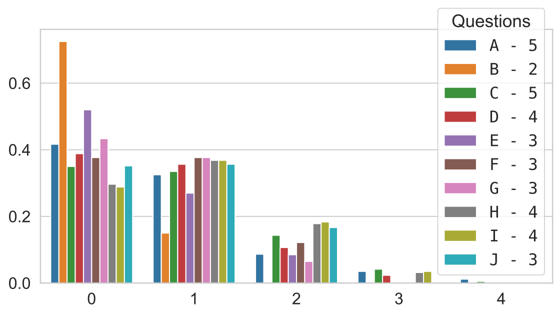

Figure 2: Distribution of answer distances between two annotators on labels for the same document-summary pair.

It can be observed that 30% to 50% of the labels show agreement between annotators, excluding answer B, where the agreement elevates to more than 70% of the cases. However, if we aggregate the annotations with agreement and the ones that are at a distance 1, we cover nearly 80% of the observations in each answer.

With the information extracted from Table 2 and Fig. 2, along with the average Cohen's Kappa between annotators in Appendix C, it can be gathered that the labeling process produced a dataset that captured subjectivity but retained enough agreement to consider the data coherent and valid, from where useful information could be extracted.

2.6 Dataset Variants

In the complete dataset, called ***Annotators***, for each document-summary pair, there are 3 samples (one per annotator). We also compiled a dataset called ***Median***, where the label for a certain question is the median of the 3 corresponding labels.

3 Abstractivity-related Questions Correlation Analysis

In this section, we analyze in the CLAsum dataset whether the answers to the questions related to the actions identified by Jing correlate with the perception of the level of abstractivity that the annotators had regarding the viewed summaries.

Fig. 3 presents Pearson's correlation of questions from C to J (abstractivity-related questions) for the *Annotators* dataset. Additionally, we introduce a new column ($\tilde{\mathbf{x}}[D..J]$), the median of the 7 aspects (D to J) normalized by the maximum value that can acquire each question.

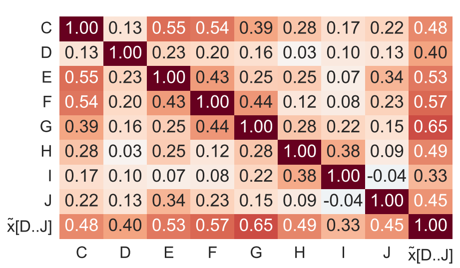

Figure 3: Pearson's correlation between two questions in *Annotators* dataset. $\tilde{\mathbf{x}}[D..J]$ is the normalized median from D to J.

Considering the first column of the matrix (C, perception of abstractivity), two main questions present a moderate correlation with the presence of abstractivity: (E) sentence information melting and (F) syntax alteration, which it is quite clear that it is necessary to create more abstractive summaries.

Using synonyms (G) and generalizations (H) show a low correlation with C, but they still relevant. When we consider the last column, which condenses the perception of the level usage of Jing's actions, it shows a moderate correlation with C, which means that the perception of abstractivity is related to how frequently those actions were used to compose a summary.

We also studied the correlations using the *Median* dataset, Fig. 4 shows the results.

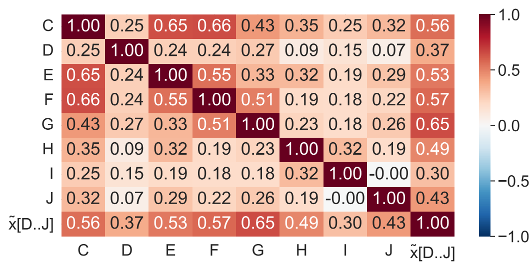

Figure 4: Pearson's correlation between two questions in *Median* dataset. $\tilde{x}[D..J]$ is the normalized median from D to J.

Questions D and C went up to a high correlation, which is understandable since more abstractive summaries tend to cover more information by joining the information from more sentences, requiring more syntactic changes. The rest of the actions fluctuate between low and moderate correlation. Regarding the last column, the correlation went up closely to the range of high correlation but remained moderate.

All these observations confirm our hypotheses that Jing's actions are related to the level of abstractivity perception, and, that there exists a single continuous dimension where the two styles, extractive and abstractive summarization, could coexist.

4 Abstractivity Characterization

Based on the conclusions of Section 3, we identify two ways of describing abstractivity in summaries: (1) how often the actions for re-writing and synthesizing the main information from a text have been used to include the information in the summary, and (2) identifying the perception of paraphrasing of the main information of a text included in the summary. This leads us to define two novel tasks regarding the abstractivity in summaries:

(1) Abstractivity Inducting Features (AIFs) extraction: Given a document and a summary, the system describes a set of 7 actions in different grades. (1) Content exclusion [a value from 1 to 4], (2) Sentence information melting [1 to 3], (3) Syntax alteration [1 to 3], (4) Synonym usage [1 to 3], (5) Generalization usage [1 to 4], (6) Specification usage [1 to 4], and (7) Content reordering [1 to 3]. Higher values in a feature indicate a wider presence of a certain action in the summary composition.

(2) Abstractivity Level (AL) prediction: Given a document and a summary, the system predicts the level of perception of how much the structure of the document's main content has been modified to be included in the summary. A value from 1 to 5, where 1 indicates an extractive summarization style and 5 indicates a strong perception that the summary's author has created it with "their own words".

The two posed tasks can be approached as ordinal classification or regression problems since both tasks were designed using Likert scales.

5 Experimentation

In this section, we detail the experimentation done with both *Median* and *Annotators* datasets. The experimental work has a double objective; on the one hand, we want to validate our perception that extractivity and complete abstractivity are extreme points of a single dimension with multiple abstractivity levels, and on the other hand, we want to verify the role of AIFs in the characterization of these levels of abstractivity.

5.1 Supervised Machine Learning Methods

To tackle the Abstractivity Inducting Features and Level classification/regression tasks, we selected a wide range of classical supervised machine learning methods, all of which were approached with the implementation available in the Scikit-Learn (Pedregosa et al., 2011) Python module.

For classification, the methods that were considered are the following: *Logistic Regression*, *Linear SVM*, *SVM with RFG kernel*, *Random Forest*, and *Multi-Layer Perceptron*. For regression, we used same methods, but *Linear Regression* instead of *Logistic Regression*.

Since some of the methods can not handle more than one feature in the output, we circumvented this handicap by training one model for each feature in the case of AIFs tasks.

5.2 Feature Extraction

Ahuir previously shown (Ahuir et al., 2021) that combining a set of abstractivity-related metrics (such as *Coverage*, *Density*, *Content Reordering*, *Abstractivity (p=[2,3])*, *Novel [2,3,4]-grams*) is useful for abstractivity measurement, and we did not want to include additional variables in the study, then, we followed the same feature extraction as in Section 2.1.

This representation of the document-summary pairs was a straightforward first approach for abstractivity-related tasks. Exploring the impact of other kinds of feature extraction methods, such as incontextual or contextual embeddings, would be out of the scope of the present work.

5.3 Evaluation Metrics

We selected a set of metrics for both the classification and regression approaches. For the classification approach, the macro versions of *Precision*, *Recall*, and *F1-score* were used; for the regression approach, we employed *Root Squared Mean Error* (RMSE), *Median Absolute Error* (MdAE).

Additionally, for classification and regression, the *Relative* distance (Eq. (1), Section 2.5) was used in the Abstractivity Level tasks, and the *Minkowski* (p=7) distance to measure the distance between the AIFs prediction vector and the reference vector since we want to evaluate the extracted features' cohesion. The *Minkowski* distance between vectors was measured against the normalized AIFs vectors. The normalized AIFs vectors with values from [0, 1] were obtained by dividing each aspect by the maximum value possible for that aspect.

Minimizing *Relative* distance (Abstractivity Level prediction) and *Minkowski* distance (AIFs extraction) will be the main goal to achieve since we want our systems to be as close to the real prediction as possible, and these metrics reflect that need.

5.4 System Types Developed

We developed two systems *End-to-End* for each proposed task: one for AIFs classification, another for the regression version of that task, and another two for Abstractivity Level classification and regression. Thus, given a document and a summary, the system first extracts the selected features representing the document-summary pair (Section 5.2), and then performs the classification/regression tasks.

Additionally, we developed a third model type (*AIFs-to-AL*), which receives the document-summary features plus the AIFs as the input and predicts the Abstractivity Level. The model was trained with the reference AIFs labels as input, along with the document-summary features. This model type should help to analyze how beneficial the inclusion of the AIFs is in the Abstractivity Level prediction.

With the *AIFs-to-AL* models, we created a *Pipeline* for Abstractivity Level prediction. The *Pipeline* receives a document-summary pair, extracts the document-summary features, and with them, the AIFs predictor extracts the corresponding AIFs. Finally, the AIFs are concatenated with the document-summary features and passed to the *AIFs-to-AL* model to obtain the prediction of the Abstractivity Level. The *Pipeline* should help verify the usefulness (for Abstractivity Level prediction task) of a system that considers the AIFs' information compared to a system that does not use them (the *End-to-End* systems).

5.5 Training and Evaluation Methodology

With only 175 document-summary pairs to work with, we were facing a low-data situation. For this reason, we trained and evaluated all system configurations 20 times with different partitions, which will show the variability in the performance of each configuration and the conclusions extracted from the results would not be tied to any random aspect of the validation process.

Considering that the distribution of the classes regarding the abstractivity level is not well-balanced, we did not use the K-Fold methodology. Instead, we split the dataset with a different random state (seed) each time. In each partition, 20% of the document-summary pairs were put aside for testing and the rest for training. The partitions were created with the `train_test_split` from Scikit-Learn, setting the seed with an integer number from 0 to 19 and stratified with the C answer (abstractivity level) from *Median* dataset. We verified that all train partitions contain all the possible labels/answers for each question and that 99.4% of the samples were used for testing at least once. Also, it should be mentioned that, in the *Annotators* dataset, all samples that contain the same document-summary pair were placed in the same partition (test or train).

Regarding the sample distribution for training and testing, it should be noted that only one sample per document-summary was available for classification (*Median* dataset). However, for regression (*Annotators* dataset) three samples per document-summary were available, which aimed to capture the diversity obtained by the labeling process.

For the configuration of each Supervised Machine Learning Method, we bypass modifying the default parameters of the Scikit-Learn implementation (version 1.5.0) to avoid introducing more variables in the study. Only the random state was set to 42 when the method had this feature and increased the max steps to 1 000 000 (a limit that was never reached).

6 Systems' Results

This section presents the results obtained by the best system configurations for the two Abstractivity tasks in classification and regression approaches.

All tables follow the same structure. There is a column for each metric, and at the right side of each name, there is an up arrow (↑) indicating that a higher value indicates better performance, or a down arrow (↓) if lower is better. In each table's numeric cell, the average value and the 95% confidence interval (exponent = lower bound, subscript = upper bound) are shown.

The names of the configurations were shorted for the sake of clearance. The short name are LiR (Linear Regression), LgR (Logistic Regression), lSVM (Linear SVM), Multi-Layer Perceptron (MLP), RnF (Random Forest), and SVM (SVM with RFG kernel). Also, some title names of the columns were shorted: *Mthd* (Method), *M-Dist* (Minkowski distance), *R-Dist* (Relative distance), *Precis* (Precision), and *MdAE* (Median Average Error).

6.1 Abstractivity Inducing Features tasks

In this section, the best results for AIFs extraction tasks when we consider Mikowski distance (*M-Dist*) as the reference metric are shown. Extended table results are in Appendix D.

Table 3 shows the best results for the classification task, the *Random Forest* model.

Mthd	M-Dist ↓	Precis ↑	Recall ↑	F1 ↑
RnF	$35.5^{34.7}_{36.3}$	$47.9^{45.0}_{50.9}$	$45.4^{43.6}_{47.3}$	$43.4^{41.5}_{45.2}$

Table 3: Results of the best model for Abstractivity Inducing Features classification task in *Median* dataset.

Regarding the *M-Dist* average results, we can extract the AIFs predicted vectors average a distance of 36% of the reference vector. The predictions should be considered close enough to be useful, considering that the distance is from a comparison of 7-sized vectors with at least 3 values per feature.

Table 4 shows the best results for AIFs extraction in *Annotators* dataset, the Multi-Layer Perceptron model.

Mthd	M-Dist ↓	RMSE ↓	MdAE ↓
MLP	$38.9^{38.4}_{39.4}$	$0.76^{0.75}_{0.77}$	$0.57^{0.56}_{0.58}$

Table 4: Results of the best model for Abstractivity Inducing Features regression task in *Annotators* dataset.

Relevant results have been achieved for regression. If we consider *RMSE* or *MdAE*, it is noticeable that the model averages less than one level of difference between the predicted feature and the reference one, which indicates that the model can infer a helpful AIFs vector from the abstractivity indicators.

6.2 Abstractivity Level tasks

This section presents the best results for each type of system for the Abstractivity Level tasks. The first type is the *End-to-End* (E), the second one is the *Pipeline* (P), and the third model is *AIFs-to-AL* (A). Due to that *AIFs-to-AL* uses the reference AIFs vectors, it can be considered an upper bound of the *Pipeline*.

Table 5 shows the best systems for the *Median* dataset.

Type	Mthd	R-Dist ↓	Precis ↑	Recall ↑	F1 ↑
E	RnF	$10.6^{09.7}_{11.5}$	$64.2^{60.7}_{67.6}$	$60.0^{56.4}_{63.6}$	$60.1^{56.8}_{63.5}$
P	SVM+RnF	$\mathbf{10.2}^{09.2}_{11.2}$	$\mathbf{64.4}^{60.3}_{68.5}$	$\mathbf{60.3}^{56.8}_{63.8}$	$\mathbf{60.3}^{56.7}_{64.0}$
A	RnF	$9.6^{08.6}_{10.6}$	$68.5^{64.6}_{72.4}$	$63.1^{59.1}_{67.0}$	$63.5^{59.6}_{67.5}$

Table 5: Results of the best system per system type for Abstractivity Level classification task in *Median* dataset.

Results show that the *Pipeline* system performs slightly better than the *End-to-End* system. This indicates that the AIFs information has positively influenced the performance of the classification task, and it could be improved further if we consider the *AIFs-to-AL* model type scores. However, the *Pipeline* lost 5% of performance due to the cumulated error associated with the AIFs predictor model.

Regarding the regression results, Table 6 shows the results of the best model for each type.

Type	Mthd	R-Dist ↓	RMSE ↓	MdAE ↓
E	lSVM	$14.9^{14.2}_{15.5}$	$0.96^{0.93}_{1.00}$	$\mathbf{0.57}^{0.54}_{0.61}$
P	SVM+lSVM	$\mathbf{14.7}^{14.1}_{15.4}$	$\mathbf{0.95}^{0.92}_{0.98}$	$0.59^{0.56}_{0.62}$
A	lSVM	$13.6^{13.1}_{14.1}$	$0.88^{0.85}_{0.91}$	$0.56^{0.53}_{0.59}$

Table 6: Restuls of the best system per type for Abstractivity Level regression task in *Annotators* dataset.

In regression, we observe a similar trend as in classification. The information from the AIFs was beneficial for abstractivity level prediction. However, the impact of the AIFs predictor was more noticeable than in classification if we consider the difference in *Pipeline* performance and the *AIFs-to-AL* model.

Since we have observed that AIFs information benefits Abstractivity Level prediction, we compare the confusion matrices (CM) for the 20 runs and 35 test samples per run (700 samples). The number in the `y-axis` is the reference label, and the one in the `x-axis` is the predicted. Numbers in cells indicate the number of samples in each combination. Colors indicate the percentage of samples in the combination regarding the total of samples in each row (real label).

Fig. 5 shows the CMs of *End-to-End* model (a), and *Pipeline* model (b).

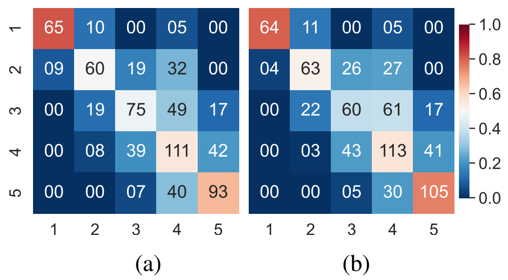

Figure 5: Confusion matrices of *End-to-End* (a) and *Pipeline* (b) in the *Median* dataset for Abstractive Level classification.

Firstly, it is noticed that both systems could correctly classify a sensible number of samples on each level, confirming that models can capture humans' Abstractivity Level perception in summaries. When we compare both systems, (a) and (b), we notice that levels of abstractivity 2, 4, and 5 have increased the number of correct samples (diagonal). Also, the number of samples mislabeled by more than one level has been reduced in levels 4 and 5. However, in level 3, the (b) model has reduced the number of correct samples, misleading level 3 with level 4.

When we compare the CMs of *End-to-End* and *AIFs-to-AL* models in Fig. 6, we obtain similar conclusions.

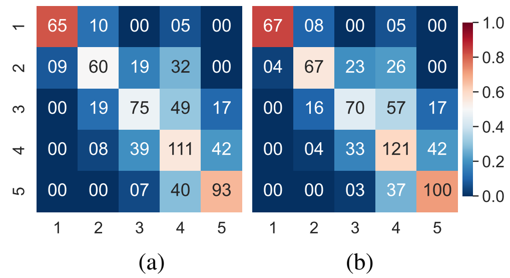

Figure 6: Confusion matrices of *End-to-End* (a) and *AIFs-to-AL* (b) in the *Median* dataset for Abstractive Level classification.

We observe that model (b) increased the number of hits in level 1, in addition to levels 2, 4, and 5. Also, the number of no-hits further than 1 level has been reduced even more than in the *Pipeline* (Fig. 5.b). Finally, the impact on level 3 was less prominent than in the *Pipeline*, but still has lower performance than *End-to-End* for this level.

Generally speaking, we can conclude that AIFs have provided useful information to improve Abstractivity Level prediction, which indicates that measuring these aspects gives additional details about how the summary was composed. Regardless of whether AIFs information was used or not, Fig. 5 shows that models could infer the Abstractivity Level for many samples, supporting the idea of a single continuous dimension where extractive and abstractive summarization coexist.

7 Conclusions

In this work, we have presented and made available to the scientific community the CLAsum dataset. This is a hand-annotated dataset that allows characterizing the complexity of the process of summarizing a document by measuring the Abstractivity Level and seven Abstractivity Inducting Features.

The results from the study of the dataset and the experimental work show how the Abstractivity Level and AIFs are related and how AIFs are useful when measuring the level of abstractivity of a summary. Our study places extractivity and complete abstractivity as the extreme points of a single dimension with multiple levels.

Limitations

Distribution Representativeness. The chosen datasets for the study were focused on the news field. Additionally, the selection of document-summary pairs has been restricted in the number of words (especially the document side). These restrictions have reduced the variety of topics that appeared during the annotation. Consequently, the distribution of the annotated samples could not be fully representative of other fields.

Annotators diversity. Even though 17 annotators from different degrees of studies and experience have been involved, all are from the field of Computer Science. Therefore, opinion diversity would be reduced in other fields of expertise and/or education levels. Additionally, all the annotators were not native English speakers. Even though annotators had high English reading skills, the fact of not being native speakers, in some specific situations, a little lost in comprehension of some particular details of the texts could appear.

Biases. During all the phases of the annotation process (guideline design, annotation process, and data gathering), one of the highest priorities was to avoid any influence on the outcome. However, we are mindful that there would always be a chance, even tiny, that unconscious actions or word selection could introduce biases in the outcome. In this regard, we believe that our work produced significantly unbiased data that the community could take as a foundation for future work.

Model Design Soundness. This work tested a set of configurations in the most straightforward possible way to reduce the number of study variables and presented a basic baseline for future works. Using a set of abstractivity indicators to represent the document-summary pairs for the two proposed abstractivity-related tasks was a direct approach. In this regard, using them all at once would not guarantee the best outcome possible with those indicators since there might be duplicated information. Therefore, a correlation study between them and the abstractivity level would help to reduce the dimensionality of the features, which could increase the performance in the tasks. The same would apply to the Abstractivity Inducting Features (AIFs), when they are joined to the rest of the indicators and used to predict the level of abstractivity in the *Pipeline* systems. Additionally, the selection of supervised machine learning methods was made without any specific criteria to guarantee the best outcome; the selection was broadly made to capture different machine learning method approximations.

Ethical Statement

Biases. The datasets from where the document-summary pairs were extracted could present biases regarding the vocabulary used or how certain topics were treated. We did not analyze the included samples in this regard; we took them randomly, as they were published in the original datasets. Regarding to the findings and statistics presented, they are related to annotation complexity and tied to the specific group of annotators involved in the process and should, therefore, be considered as approximate.

Intended Use and Potential Misuse. In relation to the dataset created in this work, it was created to provide the community with data for working and expanding the concept of abstractivity in summarization and new ways to characterize the aspect in summaries. Any different analyses or extrapolations extracted from that data would not be linked to the subject of this work and could raise ethical considerations.

References

Vicent Ahuir, Lluís-F. Hurtado, José Ángel González, and Encarna Segarra. 2021. NASca and NASes: Two monolingual pre-trained models for abstractive summarization in catalan and spanish. *Applied Sciences*, 11(21).

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. 2021. Capturing relations between scientific papers: An abstractive model for related work section generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6068–6077, Online. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 4830–4842, Online. Association for Computational Linguistics.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Xiyan Fu, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Changlong Sun, and Zhenglu Yang. 2021. RepSum: Unsupervised dialogue summarization based on replacement strategy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6042–6051, Online. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Hongyan Jing. 2002. Using hidden markov modeling to decompose human-written summaries. *Comput. Linguist.*, 28(4):527–543.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Potsawee Manakul and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041, Online. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807. Association for Computational Linguistics.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Roman Sirokov. 2024. pywebview: A lightweight cross-platform library to create web-based desktop guis.

Weisong Sun, Chunrong Fang, Yuchen Chen, Quanjun Zhang, Guanhong Tao, Yudu You, Tingxu Han, Yifei Ge, Yuling Hu, Bin Luo, and Zhenyu Chen. 2024. An extractive-and-abstractive framework for source code summarization. *ACM Trans. Softw. Eng. Methodol.*, 33(3).

Daniel Varab and Yumo Xu. 2023. Abstractive summarizers are excellent extractive summarizers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 330–339, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. 2021. BASS: Boosting abstractive summarization with unified semantic graph. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6052–6067, Online. Association for Computational Linguistics.

Chujie Zheng, Kunpeng Zhang, Harry Jiannan Wang, Ling Fan, and Zhe Wang. 2020. Topic-guided abstractive text summarization: a joint learning approach. *arXiv preprint arXiv:2010.10323*.

Yanyan Zou, Xingxing Zhang, Wei Lu, Furu Wei, and Ming Zhou. 2020. Pre-training for abstractive document summarization by reinstating source text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3646–3660, Online. Association for Computational Linguistics.

A Labeling Guideline

In this section, the completed guideline that was used in the labeling process is presented.

Given a newspaper article and a summary in the left side (Document/Summary tab), answer 10 questions/statements regarding the content of the article and the summary and/or the way the summary was created. The possible answers are detailed on the left side (Questions tab).

A) *The summary provides the most relevant information about the article, and the article extends it with additional details*:

- 0: Strongly Disagree.
- 1: Disagree.
- 2: Undecided.
- 3: Agree.
- 4: Strongly Agree.

B) *Regarding information contained in the summary*:

- 1: All the information in the summary can be found in the article (not necessarily in the exact words).
- 2: Almost all the information in the summary can be found in the article, but adds some additional information.
- 3: I can not consider the given summary a truly abstract. All the information provided in the summary, it is additional and can not be extracted or inferred from the article.

C) *What is your perception about how the author of the summary wrote it?*:

- 0: Does not apply; it is not a summary.
- 1: They rely entirely on the article. It is as if I was reading complete sentences highlighted in the article.
- 2: They rely heavily on the article to write the summary. It only presents slight changes in form and/or order concerning the article.
- 3: They mainly rely on the article to write the summary. Segments of the summary can be identified in the article. Still, the author alters the article's text in form and/or order.
- 4: They weakly rely on the article to write the summary and alter a lot of the article in form and/or order.
- 5: Overall, they do not rely on the article to write the summary; instead, they explain the main ideas of the article in their own words.

D) *How does the author handle non-relevant information in the article?*:

- 0: Does not apply; it is not a summary.
- 1: They discard complete sentences. No segments or words of a sentence are discarded.
- 2: They focus on mainly discarding complete sentences. Segments or words of sentence discarding is also present, but it is less often than complete sentences discarding.
- 3: They focus mainly on discarding text segments within the sentences of the article. The complete sentence discarding is absent, or it is noticeably less frequent than segment.
- 4: All information is considered relevant; they manage to cover all the information in the article and substantially reduce its length. discarding.

E) *For the creation of the summary, part of the information selected from the sentences of the article is combined to form the sentences of the summary*:

- 0: Does not apply; it is not a summary.
- 1: No sentences from the article are combined. Each sentence in the summary corresponds to the information contained by a sentence in the article.
- 2: Some sentences in the summary are created by combining the information contained by certain sentences from the article.
- 3: Most of the sentences of the summary are created by combining information from some sentences of the article. discarding.

F) *Sentences in the article that contain the information reflected in the summary have been syntactically altered for inclusion in the summary*:

- 0: Does not apply; it is not a summary.

1: No syntactic alterations exist to create the summary.

2: There are some syntactic alterations to create the summary.

3: There are many syntactic alterations to create the summary.

G) *When including sentences or segments of the article in the summary, the author replaces words or expressions with semantically equivalent ones*:

0: Does not apply; it is not a summary.

1: Never.

2: Sometimes.

3: Quite often.

H) *The summary includes generalizations of information extracted from the article. A generalization is describing one or more concepts using a less specific word (e.g., "Matthew and Amanda reappear in the new sequel of the acclaimed fiction movies of galactic adventures series" in the summary "Matthew and Amanda" could be grouped as "The main actors ...")*:

0: Does not apply; it is not a summary.

1: No information can be considered susceptible to generalization without a significant loss of information.

2: No information susceptible to generalization was generalized.

3: Less than half of the information susceptible to generalization was generalized; the rest was not generalized.

4: More than half of the information susceptible to generalization was generalized.

I) *The summary includes specifications of information extracted from the article. A specification would be to use expressions or words that make the information more specific (e.g., "The race driver has won his ninth F1 World Championship Grand Prix" in the summary "The race driver" could be detailed as "The F1 driver ...")*:

0: Does not apply; it is not a summary.

1: No information can be considered susceptible to specification.

2: No information susceptible to specification was specified.

3: At most, half of the information susceptible to specification was specified; the rest was not specified.

4: More than half of the information susceptible to specification was specified.

J) *The author of the summary rearranges the chosen information. For example, if facts A-B-C appear in the article, the author refers to them in the following order B-A-C in the summary*:

0: Does not apply; it is not a summary.

1: Never.

2: On one occasion.

3: On several occasions.

B ANONYM: Anonymized App Name

Fig. 7 presents the labeling application developed for the labeling process called *ANONYM (Anonymized App Name)*. The application was developed with Python 3 and PyWebview (Sirokov, 2024), a framework for developing GUI applications with HTML and CSS. The application would be capable of handling different labeling text tasks by just developing an HTML web page for the task needings (supports HTML with CSS and JavaScript). ANONYM is available as a Python module.

For the labeling task of this work, we split the labeling window into two parts. On the left side, the annotator could see the guidelines in English and Spanish ("Questions" and "Preguntas" tabs) and the Document and Summary to work with. On the right side, the annotator had the 10 questions to answer. Additionally, and to facilitate the labeling process, the application presented the exact Common Long Sequences between the document and the summary in different colors.

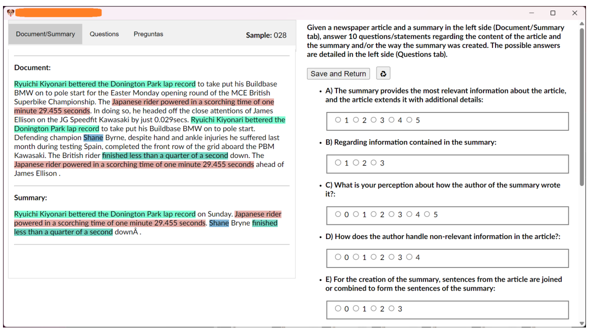

Figure 7: Labeling window of a sample in the ANONYM application.

C Average Pair-wise Annotator Agreement

Fig. 8 shows the average Cohen's Kappa agreement between two given annotators. White spaces are combinations that did not occur in the labeling process. The agreement is measured with the *Relative* distance (Eq. (1), Section 2.5) between two annotators and the 10 questions at once.

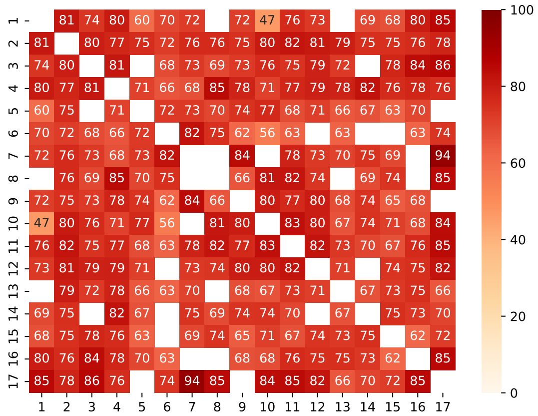

Figure 8: Average of Cohen's Kappa pair-wise agreement score (*Relative* distance).

D Extended Results for Abstractivity Inducting Features tasks

Table 7 details the results obtained by all models for the Abstractivity Inducting Features classification task with *Median* dataset. System configurations are sorted in ascending order by the *M-Dist* column.

Mthd	M-Dist ↓	Precis ↑	Recall ↑	F1 ↑
RnF	$\mathbf{35.5}^{34.7}_{36.3}$	$47.9^{45.0}_{50.9}$	$45.4^{43.6}_{47.3}$	$43.4^{41.5}_{45.2}$
SVM	$38.4^{36.9}_{39.8}$	$41.6^{38.4}_{44.7}$	$42.1^{40.7}_{43.5}$	$37.7^{36.1}_{39.3}$
LgR	$39.6^{38.6}_{40.6}$	$48.2^{45.2}_{51.3}$	$45.0^{43.4}_{46.6}$	$42.2^{40.6}_{43.9}$
lSVM	$40.7^{39.9}_{41.6}$	$46.6^{43.8}_{49.3}$	$43.7^{42.3}_{45.0}$	$40.7^{39.4}_{42.1}$
MLP	$41.9^{40.7}_{43.0}$	$\mathbf{48.5}^{46.9}_{50.2}$	$\mathbf{45.8}^{44.0}_{47.5}$	$\mathbf{44.9}^{43.3}_{46.4}$

Table 7: Results of models for Abstractivity Inducting Features classification task in *Median* dataset.

Table 8 shows the results obtained by all models for the AIFs regression task with *Annotators* dataset.

Mthd	M-Dist ↓	RMSE ↓	MdAE ↓
MLP	$\mathbf{38.9}^{38.4}_{39.4}$	$\mathbf{0.76}^{0.75}_{0.77}$	$\mathbf{0.57}^{0.56}_{0.58}$
LiR	$39.0^{38.6}_{39.5}$	$0.77^{0.75}_{0.78}$	$0.59^{0.58}_{0.60}$
lSVM	$40.6^{40.1}_{41.0}$	$0.78^{0.77}_{0.79}$	$0.59^{0.57}_{0.60}$
RnF	$40.8^{40.3}_{41.3}$	$0.79^{0.78}_{0.80}$	$0.58^{0.56}_{0.60}$
SVM	$40.9^{40.3}_{41.5}$	$0.78^{0.77}_{0.79}$	$0.55^{0.53}_{0.56}$

Table 8: Results of models for AIFs regression task in *Annotators* dataset.

E Extended Results for Abstractivity Level tasks

Table 9 details the results for Abstractivity Level classification in *Median* dataset obtained by all configurations for *AIFs-to-AL* (A), *End-to-End* (E) systems, and top-10 configurations for *Pipeline* (P) systems. Numbers in bold are the best average values in their columns, excluding type *A* type systems since they are not models that can work with the document-summary text, they need the AIFs information.

Type	Mthd	R-Dist ↓	Precis ↑	Recall ↑	F1 ↑
A	RnF	$9.6^{08.6}_{10.6}$	$68.5^{64.6}_{72.4}$	$63.1^{59.1}_{67.0}$	$63.5^{59.6}_{67.5}$
P	SVM+RnF	$\mathbf{10.2}^{09.2}_{11.2}$	$64.4^{60.3}_{68.5}$	$60.3^{56.8}_{63.8}$	$60.3^{56.7}_{64.0}$
P	RnF+LgR	$10.3^{09.5}_{11.2}$	$64.3^{60.9}_{67.7}$	$57.7^{54.3}_{61.1}$	$58.2^{54.9}_{61.5}$
P	lSVM+RnF	$10.4^{09.4}_{11.4}$	$\mathbf{64.7}^{60.6}_{68.7}$	$\mathbf{60.8}^{57.2}_{64.4}$	$\mathbf{60.6}^{56.9}_{64.3}$
P	MLP+LgR	$10.4^{09.4}_{11.4}$	$62.6^{57.9}_{67.2}$	$57.9^{53.5}_{62.3}$	$58.0^{53.8}_{62.1}$
P	MLP+RnF	$10.5^{09.5}_{11.4}$	$64.4^{60.4}_{68.3}$	$60.0^{56.7}_{63.4}$	$60.2^{56.8}_{63.6}$
P	LgR+RnF	$10.5^{09.5}_{11.5}$	$63.3^{59.5}_{67.0}$	$60.1^{56.7}_{63.5}$	$59.8^{56.3}_{63.3}$
E	RnF	$10.6^{09.7}_{11.5}$	$64.2^{60.7}_{67.6}$	$60.0^{56.4}_{63.6}$	$60.1^{56.8}_{63.5}$
E	LgR	$10.6^{09.6}_{11.6}$	$62.8^{58.4}_{67.3}$	$57.5^{53.9}_{61.1}$	$56.9^{53.4}_{60.4}$
P	RnF+RnF	$10.7^{09.8}_{11.5}$	$64.3^{61.0}_{67.7}$	$59.4^{55.9}_{62.9}$	$59.8^{56.5}_{63.1}$
P	LgR+LgR	$11.0^{10.0}_{12.0}$	$63.1^{59.0}_{67.2}$	$58.2^{54.6}_{61.8}$	$57.7^{54.1}_{61.3}$
P	lSVM+LgR	$11.0^{10.0}_{12.1}$	$62.6^{58.2}_{67.1}$	$57.5^{53.9}_{61.2}$	$57.3^{53.5}_{61.1}$
A	LgR	$11.1^{10.2}_{11.9}$	$58.5^{54.7}_{62.3}$	$54.5^{51.5}_{57.6}$	$54.9^{51.7}_{58.1}$
P	RnF+lSVM	$11.1^{10.1}_{12.1}$	$61.1^{56.7}_{65.5}$	$56.4^{52.5}_{60.3}$	$55.8^{51.9}_{59.6}$
E	lSVM	$11.3^{10.4}_{12.1}$	$61.5^{56.6}_{66.4}$	$55.8^{52.6}_{59.0}$	$54.4^{50.9}_{57.8}$
A	lSVM	$11.3^{10.2}_{12.5}$	$56.8^{52.4}_{61.3}$	$54.3^{50.0}_{58.5}$	$53.9^{49.6}_{58.1}$
A	MLP	$12.0^{10.7}_{13.3}$	$58.3^{53.6}_{62.9}$	$54.3^{49.8}_{58.8}$	$54.4^{50.1}_{58.7}$
A	SVM	$12.3^{11.7}_{13.0}$	$50.1^{46.8}_{53.3}$	$45.2^{42.6}_{47.8}$	$41.6^{39.2}_{43.9}$
E	SVM	$12.9^{12.2}_{13.5}$	$49.3^{45.9}_{52.7}$	$43.7^{41.7}_{45.7}$	$39.6^{37.9}_{41.3}$
E	MLP	$13.3^{12.6}_{14.0}$	$52.1^{49.0}_{55.1}$	$50.4^{47.7}_{53.2}$	$49.7^{46.9}_{52.5}$

Table 9: Results of systems for Abstractivity Level classification task in *Median* dataset.

Table 10 details the results obtained all *AIFs-to-AL* and *End-to-End* systems for Abstractivity Level regression task with *Annotators* dataset, and top-10 configurations for *Pipeline* systems.

Type	Mthd	R-Dist ↓	RMSE ↓	MdAE ↓
A	lSVM	13.62 $^{13.13}_{14.11}$	0.88 $^{00.85}_{00.91}$	0.56 $^{00.53}_{00.59}$
A	SVM	13.77 $^{13.30}_{14.23}$	0.89 $^{00.86}_{00.91}$	0.56 $^{00.54}_{00.59}$
A	MLP	13.83 $^{13.35}_{14.31}$	0.88 $^{00.86}_{00.91}$	0.56 $^{00.54}_{00.61}$
A	LiR	13.96 $^{13.37}_{14.56}$	0.89 $^{00.85}_{00.93}$	0.60 $^{00.56}_{00.63}$
A	RnF	14.30 $^{13.75}_{14.85}$	0.90 $^{00.87}_{00.93}$	0.60 $^{00.56}_{00.65}$
P	SVM+lSVM	**14.73** $^{14.09}_{15.37}$	0.95 $^{00.92}_{00.98}$	0.59 $^{00.56}_{00.62}$
E	lSVM	14.85 $^{14.20}_{15.50}$	0.96 $^{00.93}_{01.00}$	**0.56** $^{00.54}_{00.61}$
P	lSVM+lSVM	14.88 $^{14.27}_{15.49}$	0.95 $^{00.92}_{00.98}$	0.60 $^{00.56}_{00.63}$
P	MLP+lSVM	14.97 $^{14.43}_{15.50}$	**0.94** $^{00.91}_{00.97}$	0.64 $^{00.61}_{00.67}$
P	LiR+lSVM	14.97 $^{14.41}_{15.52}$	**0.94** $^{00.91}_{00.97}$	0.63 $^{00.60}_{00.66}$
P	lSVM+SVM	14.99 $^{14.36}_{15.62}$	0.98 $^{00.94}_{01.01}$	**0.56** $^{00.55}_{00.60}$
P	LiR+SVM	15.02 $^{14.42}_{15.62}$	0.97 $^{00.93}_{01.00}$	0.59 $^{00.56}_{00.62}$
P	RnF+SVM	15.06 $^{14.47}_{15.65}$	0.97 $^{00.94}_{01.00}$	0.62 $^{00.59}_{00.65}$
P	SVM+SVM	15.08 $^{14.45}_{15.70}$	0.98 $^{00.95}_{01.02}$	0.59 $^{00.56}_{00.62}$
P	RnF+lSVM	15.09 $^{14.53}_{15.64}$	0.95 $^{00.92}_{00.98}$	0.66 $^{00.63}_{00.69}$
P	MLP+SVM	15.09 $^{14.53}_{15.65}$	0.97 $^{00.93}_{01.00}$	0.59 $^{00.56}_{00.62}$
E	SVM	15.13 $^{14.55}_{15.72}$	0.97 $^{00.94}_{01.01}$	0.61 $^{00.59}_{00.64}$
E	MLP	15.31 $^{14.84}_{15.78}$	0.94 $^{00.92}_{00.97}$	0.66 $^{00.64}_{00.69}$
E	LiR	15.40 $^{14.76}_{16.03}$	0.96 $^{00.91}_{01.00}$	0.65 $^{00.61}_{00.70}$
E	RnF	15.45 $^{14.91}_{16.00}$	0.96 $^{00.93}_{00.99}$	0.65 $^{00.62}_{00.68}$

Table 10: Results systems for Abstractivity Level regression task in *Annotators* dataset.