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 purelly 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 with 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 with 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.

Jing (2002) conducted a study on the actions that abstraction professionals used to create their abstractive summaries. Specifically, he identified the following six actions: *sentence reduction*, *sentence combination*, *syntactic transformation*, *lexical paraphrase*, *generalization/specification*, and *content 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 add new characteristics for summaries by measuring the presence of each action identified by Jing (2002); we hypothesize that these characteristics will help to measure the level of abstractivity.

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 (2002)'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 considered: Abstractivity Inducting Features extraction, and Abstractivity Level prediction.

(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

To build a dataset that could contain diversity regarding abstractivity, we took as document-summary pairs source the test partitions of two well-known datasets in the summarization area: CNN/DailyMail (See et al., 2017), and XSum (Narayan et al., 2018). We chose these datasets since they have been deeply used with extractive and abstractive summarization methods (Giarelis et al., 2023; Alami Merrouni et al., 2023; Kabeer and Khan, 2024; Ahuir et al., 2024). We could coarsely infer what kind of summaries would be found in each dataset concerning the extractivity/abstractivity. Specifically, in the CNN/DailyMail dataset, summaries tend to be more extractive than abstractive; whereas, the ones in XSum are more abstractive than extractive; this makes both datasets complementary for this work.

For pair sampling from each dataset, we could randomly select the pairs; however, this would preserve the original distributions of each dataset. Since we aimed to create a dataset with diversity regarding abstractivity, we designed a sampling method based on clustering that could obtain pairs with summaries with different degrees of extractivity/abstractivity, independently of the original distribution of the dataset. For each dataset, 5 clusters were created using the KMeans algorithm and a set of features related to abstractivity to characterize each pair. Ahuir et al. (2021) shown that combining a set of abstractivity-related metrics is useful for abstractivity measurement; therefore, we used the following ones: 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). Consequently, an 8-component features vector was used to characterize a pair.

From the 10 clusters, 5 per source dataset, we extracted 20 document-summary pairs per cluster. The final dataset comprised 100 pairs from CNN/DailyMail and another 100 from XSum. Additionally, we set some restrictions in the selected pairs: (1) the document should contain a maximum of 500 words, (2) the summary should contain a minimum of 38 words, and (3) the proportion of words the document/summary should be at least 2:1.

With the sampling method and the filters, we ensured that the final dataset should contain summaries with different degrees of extractivity/abstractivity, which makes the dataset useful for abstractivity studies.

2.2 Labeling Guideline Design

Since our main objective was to evaluate how the information in a document was modified (removing content, merging sentences, etc.) to buid the corresponding summary, the process of labeling is based on the assumption that the analyzed texts are actual summaries. However, we noticed that the datasets contained samples with no actual summaries (the summary can not be entailed from the document), which was also perceived by other authors (Guo et al., 2022). To detect those *not-summaries*, the guideline started with two questions related to the quality of the summary: one about the information included in the summary with respect to the document **(A)** and another about the amount of new information added by the summary **(B)**. Then 8 abstractivity-related questions complete the guideline, from **(C)** to **(J)**. One question about the perception of abstractivity (question **(C)**) and 7 questions for the actions identified by Jing (2002) (from **(D)** to **(J)**); Generalization **(H)** and Specification actions **(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)** Reordering (3). In question **(C)**, 1 is the definition of extractivity summarization and 5 is the definition of a high-level abstractivity summarization. From questions **(D)** to **(J)**, higher values imply a higher presence of the corresponding action.

Aditionally, 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, 3 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.

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

The labeling process was largely unguided for annotators since half were NLP experts. We exploit this fact to verify the understandability (verified by all the annotators) and correctness (verified by the experts) of the guideline without introducing any bias. To verify the guideline, we had an initial meeting to explain the labeling pipeline and resolve any doubts about the guideline; however, no labeling examples were given. Then, annotators labeled a small subset of pairs, which helped to raise some caveats that were corrected. In this way, we obtained the final version of the guideline and ensured its quality.

The annotation process was carried out with an in-house developed labeling application called ANONYM[1] (technical details in Appendix B). ANONYM showed the guideline and document-summary on the left and the answer page on the right. Additionally, aiming to help annotators, the application showed the longest common extractive segments between the document and the summary in different colors (Fig. 6, Appendix B). With ANONYM we had a simple workflow: (1) the annotator placed the assigned pairs in a folder, (2) they annotated each pair, (3) the labels were packed, and (4) uploaded to a shared drive. This workflow also ensured that an annotator could not see labels from other annotators.

Finally, it is worth mentioning that the annotation process was especially difficult, since most

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

of the questions need some text analysis to be answered. On average, each pair took 15 minutes, which implies 150 hours of working time for creating the *CLAsum* dataset composed of 600 samples; nearly 9 hours per annotator that labeled around 36 pairs each.

2.4 Sample Distribution

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

Summary	Not a Summary
175	25

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

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 only used the subset of samples with valid summaries for the study. We refer to that subset as *CLAsum*^{s}, which contains 525 samples (175 document-summary pairs with 3 different sets of labels each).

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).

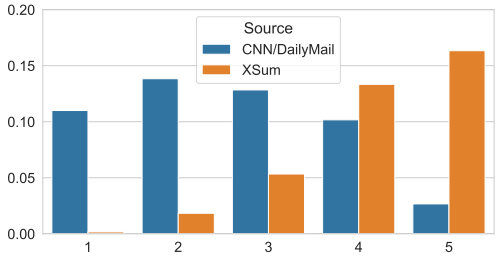

Figure 1: Distribution of answers for question **(C)** in *CLAsum*^{s}, regarding the perception of the abstractivity level in the summary.

Fig. 1 shows that although the sampling process was the same for the two sources, the achived distribution of samples differs from one source to the other, regarding perception of abstractivity. It can be noticed that the clustering process collected more diverse samples from CNN/DailyMail than from XSum, regarding the abstractivity level. XSum samples tended to present a higher abstractivity level; which verifies our initial coarse inference stated in Section 2.1.

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.

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

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 and Fleiss' Kappa for the measurement, with the *Relative* distance (Eq. (1)) as distance function between observations. *Cohen's Kappa* column is calculated with the pair-wise average score among the three annotators.

Question	Cohen's Kappa	Fleiss' 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 in per Question in the *CLAsum*^{s} subset using the *Relative* distance.

It can be observed that the average agreement with Cohen's Kappa is almost perfect. However, when Fleiss' 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.

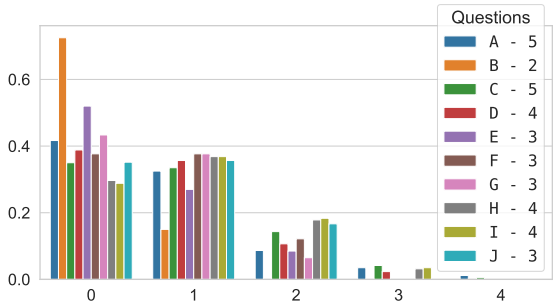

Figure 2: Distribution of answer distances between two annotators on labels for the same document-summary pair in the *CLAsum*^{s} subset.

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.

3 Abstractivity-related Questions Correlation Analysis

In this section, we analyze in the *CLAsum*^{s} subset whether the answers to the questions related to the actions identified by Jing (2002) 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). Additionally, we introduce a new column ($\mu[D..J]$), the average of the 7 questions (**(D)** to **(J)**) normalized by the maximum value that can acquire each question.

Considering the column of the *perception of abstractivity* **(C)**, two questions present a strong positive correlation with it: *sentence information melting* **(E)** and *syntax alteration* **(F)**, which indicates that they are more frequently used in the creation of more abstractive summaries. The introduction of *synonyms* **(G)** has a moderate positive correlation; hence, it has also an impact in the perception of abstractivity in summaries. The rest of the actions have a weak positive correlation with **(C)**: *excluding content* **(D)**, *include generalizations* **(H)** and *especifications* **(I)**, and *reordering content* **(J)**. It indicates that those actions have slightly more presence in higher levels of abstractivity. When we consider all Jing (2002)'s actions at the same time ($\mu[D..J]$), a strong postive correlation with **(C)** appears, with slightly higher value than the highest correlation of a single action (question **(E)**). Therefore, that indicates that the perception of abstractivity is related to how frequently those actions are used to compose a summary and how they are combined.

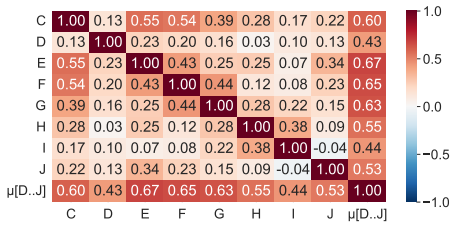

Figure 3: Pearson's correlation between two questions in the *CLAsum*^{s} subset. $\mu[D..J]$ is the normalized average from questions **(D)** to **(J)**.

Fig. 4 shows the evolution of the average answer for each question when the abstractivity level (question **(C)**) is considered. To facilitate the comparison, the answers were normalized by the maximum possible value available in their respective question. Also included the aggregation of all the answers ($\mu[D..J]$).

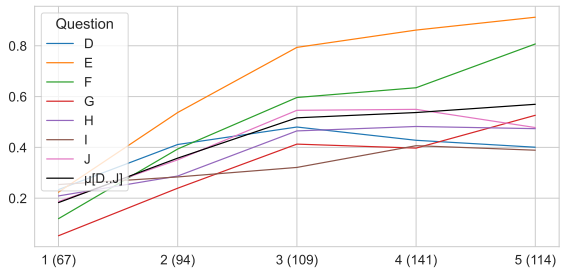

Figure 4: Average answer per question in the *CLAsum*^{s} subset. $\mu[D..J]$ is the normalized average from questions **(D)** to **(J)**. `X-axis`: level of abstractivity (question **(C)**), showing the number of samples in parethesis. `Y-axis`: average normalized answer.

From abstractivity levels 1 to 3, it can be noticed a clear positive evolution in the average answer for all the questions. Therefore, in that range of abstraction, the more often are used the actions, the higher perception of abstractivity exists. This does not occur for all the actions from abstractivity levels 3 to 5. In the case of actions *sentence information melting* **(E)** or *syntax alteration* **(F)**, they are more frequently used in higher levels of abstractivity, which indicates that more abstractive summaries tend to cover more information (*content exclusion* **(D)** decreases) by joining sentences and shortening the content with syntax alterations. Also we observe an increasing pick of usage of *synonyms* **(G)** from abstractivity levels 3 and 4 to 5; which increases the chance that the reader would feel the summary written "the author's words". In the case of actions *generalizations*, *especifications*, and *content reordering*, usage frequency does not increase in this range of abstractivity. Lastly, when we consider $\mu[D..J]$, it is noticeable that it always increases with the level of abstractivity, which sustains the obtained correlation value in Fig. 3.

All these observations corfirm our hypotheses. The level of presence of Jing (2002)'s actions are related to the level of abstractivity preception, and, that exists a single continuous dimension where the two summarization styles, extractive and abstractive, could coexist.

4 Abstractivity Characterization

Based on the conclusions of Section 3, we identify two ways, or tasks, for describing summaries in relation to extractivity/abstractivity.

The first task is the **the Abstractivity Level (AL) prediction**. This task aims to identify the deegree of perception of paraphrasing of the main information of a text included in the summary. In this work, our scale is the same as in question **(C)** (excluding answer 0). Therefore, level 1 indicates an extractive summarization style and level 5 indicates a strong perception that the summary's author has created it with "their own words".

The second task is to measure how often each Jing (2002)'s action was used for re-writing and synthesizing the main information from a text has been used to include the information in the summary. We named this task as **Abstractivity Inducting Features (AIFs) extraction**, and it should predict the answers for the 7 questions from **(D)** to **(J)** (answer 0 excluded).

5 Experimentation

In this section, we detail the experimentation done over the *CLAsum*^{s} subset. 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. Since we had 3 set of labels per sample and we wanted to capture the diversity withing each document-summary pair, both task where approached as regression tasks.

5.1 Extraction Features and Supervised Machine Learning Methods

The extraction features process was the same that was used for the clustering process described in Section 2.1. The document-summary pairs in the samples were represented by a vector of 8 abstractivity-related indicators. For the sake of clarity, we will refer to them as *abs-metrics-features*.

To tackle the Abstractivity Inducting Features extraction and Abstractivity Level prediction tasks, we selected a wide range of supervised machine learning methods that can accept the choosen freatures. All of them were used with the implementation available in the Scikit-Learn (Pedregosa et al., 2011) python module. The methods that were considered are the following: *Linear Regression*, *Linear SVM*, *SVM with RFG kernel*, *Random Forest*, and *Multi-Layer Perceptron*.

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 extraction tasks.

5.2 Evaluation Metrics

For measuring the performance of the regression tasks, we used *Root Squared Mean Error* (RMSE) and *Median Absolute Error* (MdAE). Additionally, the *Relative* distance (Eq. (1), Section 2.5) was used in the Abstractivity Level prediction task, and the *Minkowski* (p=7) distance for the AIFs extraction task. To evaluate the predictions with *Minkowski* distance, we measured the distance between the AIFs extraction 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.

5.3 Types of Architectures

For the AIFs extraction task, we used an architecture where systems receive the *abs-metrics-features* (Section 5.1) and return the AIFs vectors of the samples.

In the case of the Abstractivity Level prediction task, we used three different architectures:

(1) ***End-to-End*:** types of systems that receive the *abs-metrics-features* and returns the AL prediction for the samples.

(2) ***AIFs-to-AL*:** these types of systems receive vectors with the *abs-metrics-features* concatenated with the AIFs and predict the AL. We used the reference AIFs labels as input to train these systems, along with the *abs-metrics-features*.

(3) ***Pipeline*:** in this architecture the *abs-metrics-features* are received and it perform the AL prediction in two steps. In the first step, a system for AIFs extraction task predicts the AIFs based solely on the *abs-metrics-features*. In the second step, *abs-metrics-features* and extracted AIFs are concatenated and passed to an *AIFs-to-AL* system to perform the AL prediction task.

The *Pipeline* architecture should help verify the usefulness of the AIFs' information to perform the AL prediction task compared with the one that does not used them (*End-to-End* architecture).

5.4 Training and Evaluation Methodology

Since the *CLAsum*^{s} subset contains only 175 document-summary unique pairs, 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 (105 samples from 3 sets of labels of 35 pairs) and the rest for training (520 samples). 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 the dataset by document-summary pairs taking into account the median of the 3 answers for question **(C)**. Therefore, all the samples for a certain document-summary pair were always in the same partition. It should be also mentioned that 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.

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 set max steps to 1 000 000 (limit never reached).

6 Systems' Results

This section presents the results for both tasks, AIFs extraction and Abstractivity Level prediction.

6.1 Abstractivity Inducting Features Extraction Task

The best results for the AIFs extraction task are for the MLP system that obtained $(M{=}0.39, SD{=}0.01)$ in the Mikowski distance (*M-Dist*), $(M{=}0.76, SD{=}0.02)$ in the *RSME* metric and $(M{=}0.57, SD{=}0.02)$ in the Median Average Error (MdAE). The values in *M-Dist* indicate that AIFs extracted by systems are nearly 40% away from the reference. Results in *RSME* and *MdAE* confirm that systems extract AIFs that, component to component, tend to differ around ± 0.5 from the correct answer. These observations indicate that the *abs-metrics-features* vectors are not informative enough for extracting AIFs. We hypothesize that AIFs contain additional information concerning the abstractivity. All system results can be found in Table 4 (Appendix D).

6.2 Abstractivity Level Prediction Task

In this section, we discuss the results for each architecture for the Abstractivity Level prediction task.

Table 3 details the results for Abstractivity Level regression task obtained by the best system for each architecture: *End-to-End* (E), *Pipeline* (P), and *AIFs-to-AL* (A). Numbers in bold are the best average values in their columns, excluding *A* since these systems received the reference AIFs labels. Each cell shows the average value and the 95% confidence interval (exponent = lower bound, subscript = upper bound). The results of all considered systems are detailed in Table 5 (Appendix E).

Arch.	Mthd	R-Dist ↓	RMSE ↓	MdAE ↓
E	lSVM	$0.149^{0.142}_{0.155}$	$0.96^{0.93}_{1.00}$	$\mathbf{0.57}^{0.54}_{0.61}$
P	SVM+lSVM	$\mathbf{0.147}^{0.141}_{0.154}$	$\mathbf{0.95}^{0.92}_{0.98}$	$0.59^{0.56}_{0.62}$
A	lSVM	$0.136^{0.131}_{0.141}$	$0.88^{0.85}_{0.91}$	$0.56^{0.53}_{0.59}$

Table 3: Results of the best system per architecture for Abstractivity Level prediction task in *CLAsum*^{s} subset.

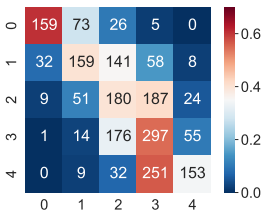

Figure 5: Confusion matrix of the best system A in the *CLAsum*^{s} subset.

We use the two-sample t-student significant test over the *R-Dist* metric to verify whether there are significant differences between them. The best *End-to-End* (E) system that takes as input the *abs-metrics-features* obatains an average performance of 0.148 evaluated with *R-Dist*, which indicates that the level of abstractivity predicted contains an of error ± 0.59 on a scale from 1 to 5. The introduction of the AIFs produce that the best *Pipeline* (P) system slightly improves the performance over the E system, $t(19)=-2.38, p<0.05$, with an average of 0.147 in *R-Dist*. However, when we consider the best *AIFs-to-AL*, that receives the reference AIFs values, the improvement over the system E is significant, $t(19)=-6.07, p<0.01$; averaging 0.136 in *R-Dist*. Therefore, we can assure that the AIFs' information helps to improve the prediction of the Abstractivity Level, though we should dispose of more reliable AIFs extraction systems. When we compare the P system with the A system, it can be concluded that the accumulated error of the AIFs extractor system used in the P architecture had a significant impact, $t(19)=5.52, p<0.01$, in reducing the effectiveness of the AIFs in the system. Note that system A is an upper bound of P systems, since A systems need AIFs in addition to *abs-metrics-features*; hence, they can not be used independently.

We calculated the confusion matrix (CM) to better understand how the system A is performing in the AL prediction task. The CMs were based on 20 runs and 105 test samples per run (2100 samples in total). 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. The color scale is related with the percentage of true positives per row. Since the systems predict the AL as a real number instead of an integer, the prediction have been rounded to compute the confusion matrix. The CM of the best system A is shown in Fig. 5.

Fig. 5 shows that the system A is capable of predicting the correct level of abstractivity of nearly half of the samples (948). Additionally, most of the samples not correctly labeled are found close to the diagonal at ± 1 level of abstractivity (966). Therefore, the system A predicts 91% of samples the correct level of abstractivity or predicts one closer to the human perception. The fact that the system can approximate close enough the level of abstractivity of the given summaries based on the *abs-metrics-features* and AIFs information, supports our idea that there is a single continuous dimension where extractivity (level 1 of abstractivity) and complete abstractivity (level 5 of abstractivity) summarization styles 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.

While this work has raised an initial frame for delving into the abstractivity aspect of summaries, several opportunities remain for future work. On the one hand, we want to extend the dataset in quantity and diversity to other different domains with the aim of allowing more generalizable studies. On the other hand, we want to study the capability of deep neural models to extract AIFs, predict abstractivity levels, and generate summaries with different abstractivity levels.

Limitations

Comprehensiveness. This study should be considered as an initial effort to delve into the understanding of abstractivity aspect of summaries since a confined amount of data was used. We collected a reliable dataset, where the 3 annotators per sample achieved a significant agreement in the annotation process, and at least one annotator per sample is an expert in the NLP field. The conclusions extracted from the study should be taken as early proof indicators of the ideas presented in this work. These ideas should be further studied with more data in different domains to be generalizable.

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, José-Ángel González, Lluís-F. Hurtado, and Encarna Segarra. 2024. Abstractive summarizers become emotional on news summarization. *Applied Sciences*, 14(2).

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).

Zakariae Alami Merrouni, Bouchra Frikh, and Brahim Ouhbi. 2023. Exabsum: a new text summarization approach for generating extractive and abstractive summaries. *Journal of Big Data*, 10(1):163.

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.

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.

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.

Nikolaos Giarelis, Charalampos Mastrokostas, and Nikos Karacapilidis. 2023. Abstractive vs. extractive summarization: An experimental review. *Applied Sciences*, 13(13).

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.

Yanzhu Guo, Chloé Clavel, Moussa Kamal Eddine, and Michalis Vazirgiannis. 2022. Questioning the validity of summarization datasets and improving their factual consistency. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5716–5727, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Adebayo Kabeer and Samiya Khan. 2024. Refining abstractive text summarization: Towards an optimizing approach based on ptlm selection and fine-tuning. In *2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6.

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. 6 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.

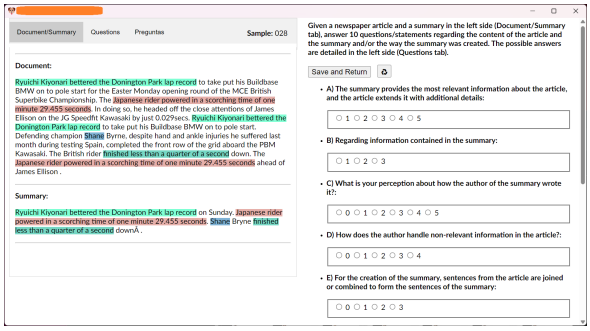

Figure 6: Labeling window of a sample in the ANONYM application.

C Average Pair-wise Annotator Agreement

Fig. 7 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.

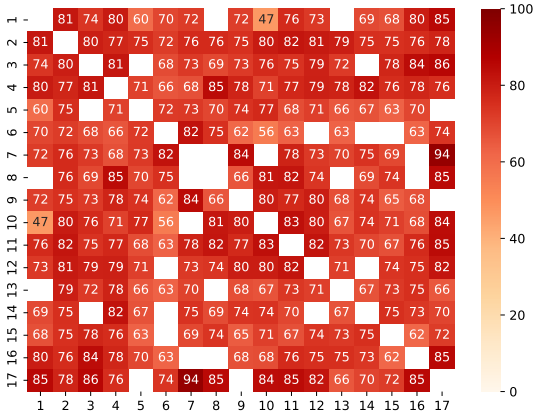

Figure 7: Average of Cohen's Kappa pair-wise agreement score (*Relative* distance).

D Results for Abstractivity Inducting Features task

Table 4 details the results obtained by all models for the Abstractivity Inducting Features extraction task in the *CLAsum*^{s} subset. System configurations are sorted in ascending order by the *M-Dist* column.

Mthd	M-Dist ↓	RMSE ↓	MdAE ↓
MLP	**0.389** $^{0.384}_{0.394}$	**0.76** $^{0.75}_{0.77}$	**0.57** $^{0.56}_{0.58}$
LiR	0.390 $^{0.386}_{0.395}$	0.77 $^{0.75}_{0.78}$	0.59 $^{0.58}_{0.60}$
lSVM	0.406 $^{0.401}_{0.410}$	0.78 $^{0.77}_{0.79}$	0.59 $^{0.57}_{0.60}$
RnF	0.408 $^{0.403}_{0.413}$	0.79 $^{0.78}_{0.80}$	0.58 $^{0.56}_{0.60}$
SVM	0.409 $^{0.403}_{0.415}$	0.78 $^{0.77}_{0.79}$	0.55 $^{0.53}_{0.56}$

Table 4: Results systems for Abstractivity Inducting Features (AIFs) extraction task in the *CLAsum*^{s} subset. The supervised machine learning methods (Mthd) are: MLP (Multi Layer Perceptron), LiR (Linear Regression), lSVM (Linear SVM), RnF (Random Forest), and SVM.

E Results for Abstractivity Level task

Table 5 details the results for Abstractivity Level prediction task in *CLAsum*^{s} subset. All the systems with architectures *AIFs-to-AL* (A) and *End-to-End* (E) are listed. In the case of systems with *Pipeline* (P) architecture, only the top-10 systems with best performance are listed. Numbers in bold are the best average values in their columns, excluding *A* since these systems received the reference AIFs labels. The table is sorted ascending by R-Dist (*Relative Distance*) column.

Arch.	Mthd	R-Dist ↓	RMSE ↓	MdAE ↓
A	lSVM	$0.136^{0.131}_{0.141}$	$0.88^{0.85}_{0.91}$	$0.56^{0.53}_{0.59}$
A	SVM	$0.138^{0.133}_{0.142}$	$0.89^{0.86}_{0.91}$	$0.56^{0.54}_{0.59}$
A	MLP	$0.138^{0.134}_{0.143}$	$0.88^{0.86}_{0.91}$	$0.56^{0.54}_{0.61}$
A	LiR	$0.140^{0.134}_{0.146}$	$0.89^{0.85}_{0.93}$	$0.60^{0.56}_{0.63}$
A	RnF	$0.143^{0.138}_{0.149}$	$0.90^{0.87}_{0.93}$	$0.60^{0.56}_{0.65}$
P	SVM+lSVM	$\mathbf{0.147}^{0.141}_{0.154}$	$0.95^{0.92}_{0.98}$	$0.59^{0.56}_{0.62}$
E	lSVM	$0.149^{0.142}_{0.155}$	$0.96^{0.93}_{1.00}$	$\mathbf{0.56}^{0.54}_{0.61}$
P	lSVM+lSVM	$0.149^{0.143}_{0.155}$	$0.95^{0.92}_{0.98}$	$0.60^{0.56}_{0.63}$
P	MLP+lSVM	$0.150^{0.144}_{0.155}$	$\mathbf{0.94}^{0.91}_{0.97}$	$0.64^{0.61}_{0.67}$
P	LiR+lSVM	$0.150^{0.144}_{0.155}$	$\mathbf{0.94}^{0.91}_{0.97}$	$0.63^{0.60}_{0.66}$
P	lSVM+SVM	$0.150^{0.144}_{0.156}$	$0.98^{0.94}_{1.01}$	$\mathbf{0.56}^{0.55}_{0.60}$
P	LiR+SVM	$0.150^{0.144}_{0.156}$	$0.97^{0.93}_{1.00}$	$0.59^{0.56}_{0.62}$
P	RnF+SVM	$0.151^{0.145}_{0.157}$	$0.97^{0.94}_{1.00}$	$0.62^{0.59}_{0.65}$
P	SVM+SVM	$0.151^{0.145}_{0.157}$	$0.98^{0.95}_{1.02}$	$0.59^{0.56}_{0.62}$
P	RnF+lSVM	$0.151^{0.145}_{0.156}$	$0.95^{0.92}_{0.98}$	$0.66^{0.63}_{0.69}$
P	MLP+SVM	$0.151^{0.145}_{0.157}$	$0.97^{0.93}_{1.00}$	$0.59^{0.56}_{0.62}$
E	SVM	$0.151^{0.146}_{0.157}$	$0.97^{0.94}_{1.01}$	$0.61^{0.59}_{0.64}$
E	MLP	$0.153^{0.148}_{0.158}$	$0.94^{0.92}_{0.97}$	$0.66^{0.64}_{0.69}$
E	LiR	$0.154^{0.148}_{0.160}$	$0.96^{0.91}_{1.00}$	$0.65^{0.61}_{0.70}$
E	RnF	$0.156^{0.149}_{0.160}$	$0.96^{0.93}_{0.99}$	$0.65^{0.62}_{0.68}$

Table 5: Results systems for Abstractivity Level prediction task in the *CLAsum*^{s} subset. The supervised machine learning methods (Mthd) are: MLP (Multi Layer Perceptron), LiR (Linear Regression), lSVM (Linear SVM), RnF (Random Forest), and SVM. For systems with *Pipeline* architecture, the Mthd format is `Sys1+Sys2`; indicating that `Sys1` is the system used for extracting the AIFs and `Sys2` the one used for predicting the Level of Abstractivity.