Evaluating story generation through automated metrics: reanalyzing the HANNA dataset

Léo Houairi (*)
ENSAE Paris, IP Paris
leo.houairi@ensae.fr

Conrad Thiounn (*)
ENSAE Paris, IP Paris
conrad.thiounn@ensae.fr

Abstract

Automatic Story Generation (ASG) is a popular branch of Natural Language Processing (NLP). As any field, in order to improve its models, it requires reliable ways to measure the quality of its outputs. Since human evaluation is costly, the development of Automatic Evaluation Metrics (AEM), that should be correlated with human judgement, is a crucial area of research. In this paper, we use the HANNA dataset to benchmark the capabilities of different AEM, reproducing some experiments of the paper that introduced this dataset [4]. Thus, our research and the structure of this paper are largely drawn from it. Our code is available on Github[1].

1 Introduction

The field of automatic story generation (ASG) has gained much attention in recent years due to its potential to create personalized and engaging content for various applications. ASG involves designing systems that can write coherent stories without human intervention. This area of research has come a long way, and current ASG systems are capable of generating stories that have well-defined plots, characters, and settings. However, despite the progress made, several challenges remain, such as creating stories that are emotionally engaging, character-driven, and have a compelling narrative arc.

One exciting development in ASG is the generation of stories associated with a particular emotion [7]. This type of ASG system aims to produce a story that evokes a specific emotional response in the reader. For example, a story generator might be designed to produce a story that elicits a sense of fear, excitement, or joy in the reader. The ability to generate emotionally engaging stories has many potential applications, such as in the entertainment industry or for educational purposes.

Another challenge in ASG is developing systems that can generate stories that are character-driven, where the actions of the characters drive the narrative. Creating such stories requires a deep understanding of human psychology and behavior, and the ability to generate characters that are both relatable and engaging. ASG systems that can produce character-driven stories have the potential to create highly personalized and engaging content for a wide range of applications, such as in gaming or interactive storytelling.

Also, a recent paper mentions the challenge of *controllability*, *i.e.* the extent to which the input controls the story itself [2]. Approaches to solve this issue are often ending-focused or storyline-focused: the system is trained so that the story has a particular end or follows a certain outline. The authors also mention the difficulty to incorporate common knowledge in ASG systems and the general question of creativity.

Other researchers have pointed out that from a single prompt, *many* different stories are possible, so the task might be underspecified. In an attempt to deal with this issue, they extracted texts written by the STORIUM online community, yielding 6000 lengthy and richly annotated stories [1]. They also propose an innovative way to evaluate the performance of their model: on the website of the community, authors can ask the model to propose a continuation for their story. Then, its quality is determined through the number of deletions and additions done by the author himself. Of course, this interesting approach is not scalable, nor applicable to different tasks.

Overall, the field of ASG is rapidly evolving,

(*) The contributions of the authors to this article are equal, therefore the order of the authors was drawn with a Bernoulli law of probability 0.5. The study only engages the authors and does not engage ENSAE Paris and INSEE.

[1] https://github.com/leohouairi/NLP-text-similarity

and future research will undoubtedly explore new approaches and techniques for generating engaging and personalized stories. As ASG systems become more advanced, they may revolutionize the way stories are created, distributed, and consumed.

In order to reach its full potential, which involves making progress regarding these different challenges, ASG requires robust metrics to evaluate the quality of its outputs. In all ways, human judgement remains the most reliable way to perform the subtle evaluation of an NLP task. But such evaluation induces a high cost, both in money and time, so many research is dedicated to the development of AEM.

Since AEM serve as a proxy for human judgment, they are usually developed so as to be highly correlated with human evaluations. However, recent research demonstrates that various AEM tend to be very correlated with one another, but poorly with human evaluations. Such result suggests that the development of new metrics should be focused on being complementary with the old ones, rather than solely on improving the correlation with human judgement [10].

Here, we focus on evaluations procedures comparing the generated text with a reference one, deemed *reference-based*[2]. When introducing new NLP systems, such measures are often presented as evidence of their quality. For example, in the paper introducing BART [14], the authors report how their model is better than previous ones in terms of `ROUGE` and `BLEU` on specific tasks.

2 Related work

2.1 A typology of metrics

Within the domain of reference-based metrics, AEM can either be *string-based*, *embedding-based* or *model-based*.

String-based metrics evaluate the similarity of two texts by analyzing the raw text through different means, notably the co-occurrences of n-grams. Famous examples of such metrics are `ROUGE` [15] and `BLEU` [17]. But this approach is limited, since it cannot take into account complexity of the language such as synonyms.

In contrast, embedding-based metrics are computed using embeddings of words, and not the words themselves. There are two types of embeddings: (i) simple word embeddings, such as those obtained through word2vec [16] where each word is linked to a unique embedding, (ii) contextualized word embeddings, such as those obtained using BERT [13], where the embedding of each word depends on its context.

Model-based metrics make use of the language representation contained within pre-trained language models.

2.2 A growing body of metrics

There are numerous AEM and it is out of the scope of this paper to describe them all. Here, we only want to highlight that the design of AEM is a fast-moving field.

Popular string-based metrics, such as `BLEU` [17] and `ROUGE` [15] are almost two decades old. Since then, many more metrics were proposed. In 2005, `METEOR` [3] was introduced: aiming at overcoming the shortcomings of `BLEU`, it was still based on the *n-gram* matching philosophy.

With the advent of embeddings, new metrics were designed to leverage the representation they provide, such as `BERTScore` (relying on BERT's embeddings) [20] and `MoverScore` (aggregates the information of different layers through a power mean) [21] in 2019, `BaryScore` in 2021 (uses Wasserstein barycenter from optimal transport theory) [8], `DepthScore` in 2022 (relies on a pseudo-metric based on data-depth) [18].

Another axis of research was also developed, where metrics rely on pre-trained models, yielding `BARTScore` in 2021 (which evaluates the similarity between texts as the probability that a *seq2seq* generates one given the other) [19] and `InfoLM` in 2022 (uses a pre-trained masked language model to represent texts) [12].

It is worth mentioning that the comparison of different systems evaluated on various tasks involves a non-trivial step of aggregation. As pointed out by [11], averaging the scores of the systems on the different tasks is not a good practice. One should rather perform the aggregation based on the *rankings* on each task, using Borda's count.

2.3 HANNA dataset

The `HANNA` dataset was released in a recent article [4]. For 96 ASG prompts, it contains one story generated by a human and 10 generated by ASG systems. It also comes with a rating of each story by 3 different human annotators, on 6 crite-

[2] Other AEM are *reference-free.*

ria[3]. The authors computed 72 AEM and studied their correlation with human evaluation.

It is important to note that the metrics we study here rate each generated text by some measure of its "proximity" with the human gold associated to the same prompt. However, when a human considers the criteria used in the HANNA dataset, he generally does not need this gold reference. Still, the annotators were given access to the human gold text to calibrate their judgement.

The goal of this paper is to reproduce part of the experiments done in [4] using a smaller number of metrics.

3 Problem Framing

3.1 Metric

Recall that we are interested in reference-based automatic metrics. We adopt the mathematical setup clearly laid out in [12], that we reproduce in the following paragraph. Formally, we consider a dataset $\mathcal{D} = \{\boldsymbol{x_i}, \{\boldsymbol{y_i^s}, h(\boldsymbol{x_i}, \boldsymbol{y_i^s})\}_{s=1}^{S}\}_{i=1}^{N}$ where $\boldsymbol{x}_i$ is the i-th reference text; $\boldsymbol{y}_i^s$ is the i-th candidate text generated by the s-th NLG system; N is the number of texts in the dataset and S the number of systems available. The vector $\boldsymbol{x_i} = (x_1, \cdots, x_M)$ is composed of M tokens and $\boldsymbol{y_i^s} = (y_1^s, \cdots, y_L^s)$ is composed of L tokens. $h(\boldsymbol{x_i}, \boldsymbol{y_i^s}) \in \mathbb{R}^+$ is the score associated by a human annotator to the candidate text $\boldsymbol{y}_i^s$ when comparing it with the reference text. We aim at evaluating an AEM f, such that $f(\boldsymbol{x_i}, \boldsymbol{y_i^s}) \in \mathbb{R}^+$.

For the HANNA dataset, N = 96 and S = 10.

3.2 Correlations

AEM are evaluated through their correlation with human judgment. Three correlation coefficients are usually used: Pearson, which measures linear relationships; Spearman, closely related to Pearson, but that is based on the rank of the observations; and Kendall, which measures how the ranking of the points is similar along the two considered axis.

Furthermore, the metrics can be evaluated at two distinct levels, depending on what exactly we try to measure: we can compute the *text-level correlation* or the *system-level* correlation. The exact mathematical formalism of these two types of correlations is available in appendix B.

[3]Relevance, coherence, empathy, surprise, engagement and complexity.

Shortly, we can either compute the correlation coefficients, then take the mean other the N texts, in which case we perform *text-level correlation*, or we can take the mean over the N texts and then compute the correlation coefficients, leading to *system-level correlation*.

4 Experimental Protocol

4.1 Dataset

This work is based on the HANNA dataset, using the csv file containing the prompts, stories and human annotations as the source file [4]. For more details, see the original paper [4].

4.2 Studied metrics

We focus on a handful of metrics, described in more detail in appendix A. String-based metrics: BLEU [17], ROUGE [15], METEOR [3]. Embedding-based metrics: BaryScore [8], DepthScore [18], BERTScore [20]. Model-based metrics: InfoLM [12], BARTScore [19]. Some scores lead to various metrics. For example, if one uses BERTScore, one can choose to rely either on the precision, recall or F1 score[5]. Thus, we study a total of 12 AEM.

4.3 Computations

The scores given to each story by the 3 human annotators were averaged. Then, the metrics listed in the precedent section were computed on each story generated by an automatic system, using as gold reference the story generated by a human for the same prompt. The correlations - the three types of them, at the two different levels - were computed between each pair of metrics, that is the 6 human ones and the 12 automatic ones. Since we were only interested in the strength of the association, only their absolute value was considered. Computation times and the source of the implementation of each AEM are available in appendix C.

Finally, we rank the AEM based on their correlations with the different human criteria, using Borda's count (see appendix D for more details).

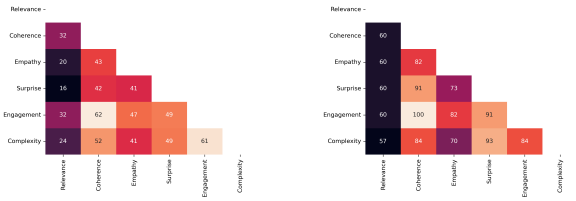
5 Results

The analysis will be focused on the Kendall correlations, figures for other types of correlations can

[4]Available at https://github.com/dig-team/hanna-benchmark-asg under the name "hanna_stories_annotations.csv".

[5]We use the notation BERTScore_P, BERTScore_R and BERTScore_F1.

be found in appendix F.



(a) Text-level (b) System-level

Figure 1: Absolute Kendall correlations (%) between human evaluations

Correlation between the human criteria. Figure 1 presents the Kendall correlation between the human criteria (see figure 4, in appendix F for a more readable version of this figure). At the text-level (1a), correlations take values between 16% and 62%. Most values are below the 50% bar, suggesting that this set of human criteria is complementary. At system-level (figure 1b), correlations are much higher: all values are superior to 50%, with one occurrence of perfect correlation; such results are in accordance with [4]. Studying Spearman's and Pearson's correlations leads to the same general observations.

Correlation between the human criteria and AEM. The results, for Kendall correlations, are presented in figures 2 and 3 at the text and system level, respectively. Once again, the text-level correlations are much lower than their system-level counterpart. In the first case, all values remain below 50% with many of them below 30%; in the second, most values are superior to 30-40 %, even if some values remain low, particularly for `InfoLM` and `BERTScore_P`. In stark contrast, at system-level, `BERTScore_R`, achieves the best correlation for 5 of 6 of the human criteria, except for empathy, where it is beaten by `BaryScore`. From this figure, `BaryScore` and `DepthScore` seem to be two other performant metrics.

As was observed in [4], most metrics are either poorly correlated with all the human criteria or highly correlated with all of them. Spearman's and Pearson's correlations yield higher values at system-levels.

Correlations between AEM. *Corresponding figures are available in appendix F.* Few general conclusions can be drawn from those correlations except for the fact that correlations are higher at the system-level. Globally, `BERTScore` seems to be weakly correlated with the other metrics, if its

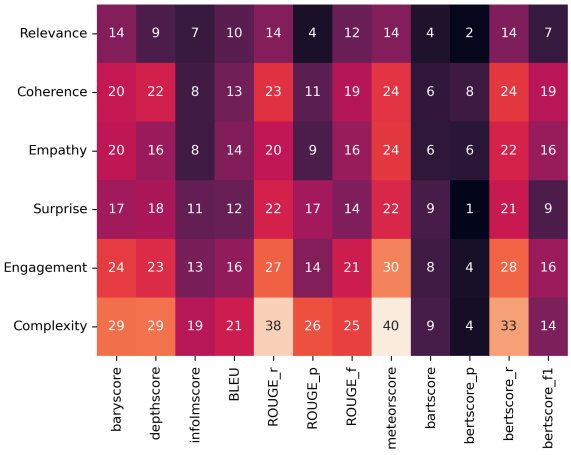


Figure 2: Absolute Kendall correlations (%) between human evaluations and automatic scores, text-level

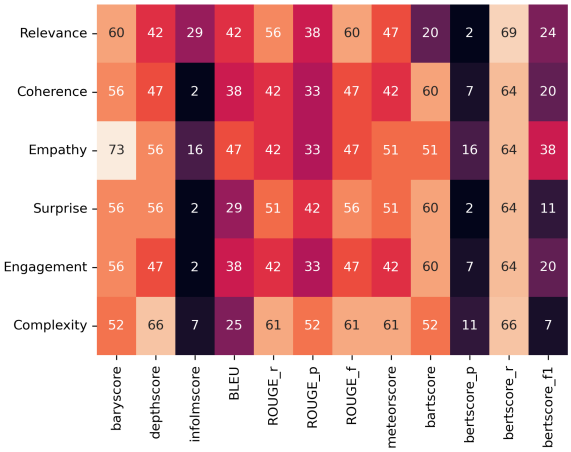


Figure 3: Absolute Kendall correlations (%) between human evaluations and automatic scores, system-level

precision of F1 score is used. In stark contrast, its recall is pretty correlated with all other AEM. The different variations of `ROUGE`, `METEOR` and `DepthScore` seem to be highly correlated, for all correlations.

Ranking the metrics using Borda's count. Table 1 presents the results of this ranking at system-level. The sum of the ranks, as well as the ranking obtained at text-level, are available in appendix E. As previously stated, `BERTScore_R`, `BaryScore` and `DepthScore` are the best AEM with Kendall correlation ranking. Spearman and Pearson correlation yield different rankings, but those three metrics are still in the top four.

However, at text-level, the three best AEM are `METEOR`, `BERTScore_R` and `ROUGE_R` (according to all correlation measures, although in different orders, see table 4).

AEM	Pearson	Kendall	Spearman
BERTScore_R	2	1	1
BaryScore	4	2	3
DepthScore	1	3	2
ROUGE_F	6	4	4
BARTScore	10	5	7
METEOR	3	6	5.5
ROUGE_R	5	7	5.5
BLEU	8.5	8	8.5
ROUGE_P	7	8	8.5
BERTSCore_F1	11	10	10
BERTScore_P	12	11	12
InfoLM	8.5	12	11

Table 1: Ranking of the metrics, system-level

6 Conclusion

In conclusion, our study highlights the importance of system-level correlations in evaluating the performance of Automatic Evaluation Metrics (AEM). While text-level correlations can also be useful, they may not provide a complete picture of the overall performance of an ASG system.

Future work in this area should consider the use of additional metrics [5, 18], such as coherence or plot structure, to complement the existing AEM. Additionally, as ASG continues to advance, it will be essential to evaluate systems on a wider range of tasks beyond the traditional story generation [9, 6]. Tasks such as generating dialogue [7] or poetry pose different challenges and may require the development of new evaluation metrics.

Finally, as noted in our study, the choice of dataset plays a significant role in evaluating the performance of ASG systems, and future work should consider the use of multiple datasets to provide a more comprehensive evaluation.

Acknowledgments

We thank Pierre Colombo for supervising this project and for helping us revise the paper. We also used one GPU during one week on the Insee's datalab.

References

[1] Nader Akoury et al. "STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation". In: *CoRR* abs/2010.01717 (2020).

[2] Amal Alabdulkarim, Siyan Li, and Xiangyu Peng. "Automatic Story Generation: Challenges and Attempts". In: *CoRR* abs/2102.12634 (2021). URL: https://arxiv.org/abs/2102.12634.

[3] Satanjeev Banerjee and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

[4] Cyril Chhun et al. "Of human criteria and automatic metrics: A benchmark of the evaluation of story generation". In: *arXiv preprint arXiv:2208.11646* (2022).

[5] Pierre Colombo. "Learning to represent and generate text using information measures". PhD thesis. (PhD thesis) Institut polytechnique de Paris, 2021.

[6] Pierre Colombo, Chloe Clavel, and Pablo Piantanida. "A Novel Estimator of Mutual Information for Learning to Disentangle Textual Representations". In: *() ACL 2021* (2021).

[7] Pierre Colombo et al. "Affect-driven dialog generation". In: *arXiv preprint arXiv:1904.02793* (2019).

[8] Pierre Colombo et al. "Automatic Text Evaluation through the Lens of Wasserstein Barycenters". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 10450–10466. DOI: 10.18653/v1/2021.emnlp-main.817. URL: https://aclanthology.org/2021.emnlp-main.817.

[9] Pierre Colombo et al. "Beam Search with Bidirectional Strategies for Neural Response Generation". In: *ICNLSP 2021* (2021).

[10] Pierre Colombo et al. *The Glass Ceiling of Automatic Evaluation in Natural Language Generation*. 2022. DOI: 10.48550/

ARXIV.2208.14585. URL: https://arxiv.org/abs/2208.14585.

[11] Pierre Colombo et al. "What are the best systems? new perspectives on nlp benchmarking". In: *arXiv preprint arXiv:2202.03799* (2022).

[12] Pierre Jean A Colombo, Chloé Clavel, and Pablo Piantanida. "Infolm: A new metric to evaluate summarization & data2text generation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 10. 2022, pp. 10554–10562.

[13] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[14] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *CoRR* abs/1910.13461 (2019). URL: http://arxiv.org/abs/1910.13461.

[15] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[16] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[17] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[18] Guillaume Staerman et al. "A pseudo-metric between probability distributions based on depth-trimmed regions". In: *arXiv preprint arXiv:2103.12711* (2021).

[19] Weizhe Yuan, Graham Neubig, and Pengfei Liu. "Bartscore: Evaluating generated text as text generation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 27263–27277.

[20] Tianyi Zhang et al. *BERTScore: Evaluating Text Generation with BERT*. 2019. DOI: 10.48550/ARXIV.1904.09675. URL: https://arxiv.org/abs/1904.09675.

[21] Wei Zhao et al. "MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance". In: *CoRR* abs/1909.02622 (2019). URL: http://arxiv.org/abs/1909.02622.

A More information regarding the different metrics

It is worth highlighting that the scales of the different metrics do not have the same signification.

In the case of metrics measuring *similarity* (precision, recall, F1 score ...), the higher the score, the better the output of the ASG system, and usually the maximum score is 1. This is the case for BLEU, ROUGE, METEOR, BERTScore and BARTScore.

On the other hand, some metrics measure a *distance* or *divergence* between an input and a reference. In this case, the lower the score, the better the output of the ASG system, and usually the lowest possible value is zero. BaryScore, DepthScore and InfoLM fall into that category.

A.1 String-based metrics

BLEU, that stands for BiLingual Evaluation Understudy is a metric proposed in 2002 for the purpose of machine translation [17]. It was meant to evaluate the quality of an automatic translation through comparison with a human translation. Shortly, it does so by computing several *n-grams* precisions. This is done by computing the number of *n-grams* present in the candidate translation as well as in the reference one, then dividing by the total number of *n-grams* in the candidate (with a slight modification: an n-gram in the reference cannot count more times than the number of times it appear in the reference). This score is called *modified n-gram precision*. Such computation can be done for various values of n: the authors propose to use the geometric mean of the of those precisions, with n up to 5.

ROUGE, standing for Recall-Oriented Understudy for Gisting Evaluation was developed following BLEU, this time for evaluating the quality of summaries [15]. The philosophy is the same: the quality of a summary is estimated through its comparison with a human one and this comparison is done on the raw text. As presented in the original paper, ROUGE is not a single score, but offers different variations along the same lines. Closely related to BLEU, ROUGE-N is a n-gram co-occurrences statistics. ROUGE-L computes a measure based on the longest common subsequence between the two texts (where a subsequence is formed by taking arbitrary words in the order of the text), ROUGE-W expands on this idea by taking into account whether the considered subsequence is made of consecutive tokens in the original text. Finally, ROUGE-S is a skip-bigram co-occurrence statistics, relying on searching common bigrams where the two tokens can be separated by other tokens.

METEOR - Metric for Evaluation of Translation with Explicit ORdering - was proposed in 2005 as a way to evaluate automatic translation [3]; its goal was to build on BLEU but to address its weaknesses. Again, it evaluates the quality of a translation by comparing it with human translations and is based on unigram-matching. But this metric considers three types of mapping: exact mapping, if the two words are the same; stem mapping, if the words share a common stem and synonym mapping, if the words are synonyms. During the computation of the metric, an *alignment* - where each unigram in a sentence is mapped to one or none unigram in the other - between the candidate and reference sentence is constructed. The construction of this alignment uses successfully each of the three types of mappings (the order matters, the first mapping considered has priority on the subsequent ones). Then, an harmonic mean is computed between the unigram precision and the unigram recall (most of the weight is on the recall). The recall is the number of unigram in the candidate translation divided by the number of unigrams in the reference text; BLEU only considered precision, while METEOR puts an emphasis on recall. Finally, this harmonic mean is penalized by a measure inversely proportional to the number of n-grams matches between the candidate and reference sentence. One weakness of this approach is that it requires an external source containing information about synonyms and stems.

A.2 Embedding-based metrics

BaryScore aims at leveraging the information contained in a multiple layer embedding (BERT in this case) based on optimal transport theory[8]. For both the candidate and reference text, and for each layer of the encoder (that creates the embeddings), a measure is constructed as a weighted sum of Dirac measures. The sum is taken over the tokens in each sentence, the weights correspond to inverse document frequencies and the Dirac mass is located at the output of the layer for the considered token. Then, separately for the candidate and reference text, the measures ob-

tained for each layer are aggregated by the computation of Wasserstein barycenters. Finally, the `BaryScore` between the candidate and reference text is the Wasserstein distance between the two Wasserstein barycenters.

`DepthScore` also uses BERT's embedding but relies on only one layer [6]. The computation of this metric involves two steps similar to those of `BaryScore`. First, for both the candidate and the reference text, a discrete probability measure is computed using one layer of BERT. Secondly, the two resulting probability measures are compared using $DR_{p,\varepsilon}$, a pseudo-metric presented in [18], that relies on the concept of data-depth to measure the dissimilarity between two distributions.

`BERTScore` relies on BERT's embedding, greedy matching and cosine similarity to compute a similarity score between a reference and a candidate [20]. First, the tokens of both sentences are passed through BERT so as to be represented by vectors embeddings. Then, the cosine similarity between all pairs of vectors embeddings is computed, and each token is matched to the token in the other sentence for which their cosine similarity is maximal (greedy matching). Given this mapping, it is possible to compute the precision (summing up the cosine similarities over the *reference* text), recall (summing up the cosine similarities over the *candidate* text) and F1 score. Any of those three can be used as metric, though the authors advise to rely on the F1 score if the task to be evaluated is machine translation. It is also possible to modify these metrics by taking into account the inverse document frequencies of the tokens.

A.3 Model-based metrics

`InfoLM` compares texts using a two-step process: (i) it computes a probability distribution over the vocabulary for both texts, (ii) then it computes a distance between those two discrete probability measures [12]. The first step involves a pre-trained masked language model (PMLM). In both the reference and candidate text, each token is masked one after the other, then the PMLM is used to compute a probability distribution for this masked token over the vocabulary. These individual distributions are aggregated through a weighted sum, where the weights correspond to inverse document frequencies. At this point, the problem boils down to comparing two discrete probability distributions. In the original paper, the authors tested various distance and divergence measures. They conclude that the *AB*-divergence leads to the best results (in terms of correlation with human judgment) but requires the tuning of two parameters. The Fisher-Rao distance also achieves good results and does not require to fine-tune any parameter.

`BARTScore` [19] relies on BART - Bidirectionial Auto-Regressive Transformer - a seq2seq pre-trained model [14]. The idea behind this metric is to evaluate the similarity between two texts by the probability of generating one given the other (which can be computed using the seq2seq model). Given a source text, an hypothesis one, and a human reference, various probabilities can be computed: the probability of the hypothesis given the source (faithfulness), of the hypothesis given the reference (precision), of the reference given the hypothesis (recall).

B Mathematical formalism of the two types of correlation

We use the formalism and explanations presented in [12] and [4], and is reproduced below. It is based on the same notations than the setting of the general problem.

The text-level correlation $C_{t,f}$ writes:

$$C_{t,f} \triangleq \frac{1}{N}\sum_{i=1}^{N} K(\boldsymbol{F}_i^t, \boldsymbol{H}_i^t)$$

where $\boldsymbol{F}_i = [f(\boldsymbol{x_i}, \boldsymbol{y_i^1}), ..., f(\boldsymbol{x_i}, \boldsymbol{y_i^S})]$ and $\boldsymbol{H}_i = [h(\boldsymbol{x_i}, \boldsymbol{y_i^1}), ..., h(\boldsymbol{x_i}, \boldsymbol{y_i^S})]$ are the vectors composed of scores assigned by the automatic metric f and the human metric (h) respectively and $K : \mathbb{R}^N \times \mathbb{R}^N \to [-1, 1]$ is the chosen correlation measure.

Similarly, the system level correlation $C_{sy,f}$ writes:

$$C_{sy,f} \triangleq K(\boldsymbol{F}^{sy}, \boldsymbol{H}^{sy})$$

$$\boldsymbol{F}^{sy} = \left[\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x_i}, \boldsymbol{y_i^1}), ..., \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x_i}, \boldsymbol{y_i^S})\right]$$

$$\boldsymbol{H}^{sy} = \left[\frac{1}{N}\sum_{i=1}^{N} h(\boldsymbol{x_i}, \boldsymbol{y_i^1}), ..., \frac{1}{N}\sum_{i=1}^{N} h(\boldsymbol{x_i}, \boldsymbol{y_i^S})\right]$$

Where the latter are the vectors composed of the averaged scores assigned by the automatic metric f and the human annotation h.

[6] The paper describing the metric is not yet published, but the metric itself is already available at https://github.com/PierreColombo/nlg_eval_via_simi_measures.

C Sources for AEM and computation times

Here we list the source of the implementation we used for each AEM.

For `BaryScore`, `DepthScore` and `InfoLM`: https://github.com/PierreColombo/nlg_eval_via_simi_measures.

For `BLEU` and `METEOR`: nltk.translate python package.

For `ROUGE`: https://github.com/pltrdy/rouge.

For `BERTScore`: https://github.com/Tiiiger/bert_score

For `BARTScore`: https://github.com/neulab/BARTScore

Regarding computation times, `BARTScore` was the longest metrics to compute (about 2 days for the whole dataset), followed by `InfoLM` (about 8 hours for the whole dataset). The other metrics required between 10 minutes and one hour.

Table 2 presents example of computation times in seconds. Each column gives the time necessary to compute the AEM on k examples, since the computation time was not strictly linear on the number of items to process.

AEM	1	1	5
`BaryScore`	9.31	9.57	63.34
`DepthScore`	9.71	8.86	48.00
`InfoLM`	22.94	22.90	161.04
`BLEU`	0.03	0.012	0.05
`ROUGE`	0.07	0.06	0.40
`METEOR`	0.20	0.032	0.28
`BARTScore`	76.00	74.54	814.036
`BERTScore`	5.32	5.56	26.42

Table 2: Example of computation times (seconds)

D Borda's count: computation

Borda's count - as an approximation of Kemeny consensus - was proposed as a way to aggregate the ranking of several systems over several tasks [11]. Here, we used it in a slightly different way: for each human criteria, we rank the AEM according to their correlation with such criteria, then use Borda's count to aggregate the 6 rankings.

We applied the following algorithm :

- For each human criteria, rank the system: the best one, the one associated with the highest correlation, gets ranking 1, the second 2 ...
- For each AEM, sum the 6 ranks it obtained.
- Rank the sum of the ranks, and use it as the final ranking. According to this procedure, the best AEM is the one with the lowest sum of ranks.

We followed this procedure at both text- and system-level, for the three different types of correlations. The tables present: (i) the ranking of each AEM (ii) the sum of the ranks of each AEM. The lines are sorted according to the rank obtained using Kendall correlation.

E Borda's count: other tables

AEM	Pearson	Kendall	Spearman
BERTScore_R	16	7.5	8
BaryScore	24	20.5	19.5
DepthScore	15	24	18.5
ROUGE_F	33	26	23.5
BARTScore	52	28.5	37.5
METEOR	19	33	35
ROUGE_R	30	35.5	35
BLEU	51	47	47
ROUGE_P	40	51	47
BERTSCore_F1	66	60.5	62
BERTScore_P	71	67	69.5
InfoLM	51	67.5	65.5

Table 3: Sum of the ranks of the metrics, system-level

AEM	Pearson	Kendall	Spearman
METEOR	1	1	1
BERTScore_R	3	2	2
ROUGE_R	2	3	3
BaryScore	6	4	4
DepthScore	5	5	5
ROUGE_F	4	6	6
BLEU	10	7	8
BERTSCore_F1	9	8	7
ROUGE_P	7.5	9	9
InfoLM	7.5	10	10
BARTScore	11	11	11
BERTScore_P	12	12	12

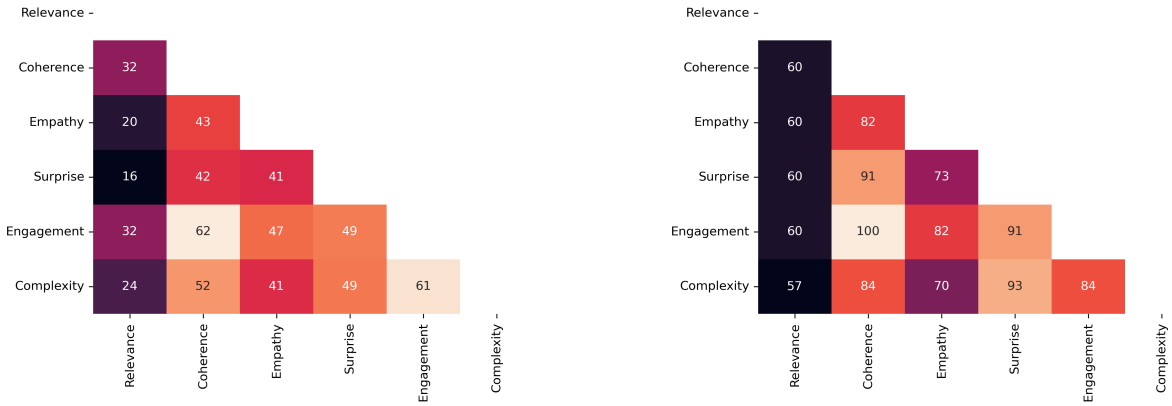
Table 4: Ranking of the metrics, text-level

AEM	Pearson	Kendall	Spearman
METEOR	9	8	8
BERTScore_R	15	15	12
ROUGE_R	13	16	18
BaryScore	32	25	23
DepthScore	31	29	33
ROUGE_F	26	39	38
BLEU	59	46	47
BERTSCore_F1	50	48	46
ROUGE_P	49	49	50
InfoLM	49	56	56
BARTScore	65	66	66
BERTScore_P	70	71	71

Table 5: Sum of the ranks of the metrics, text-level

F Other figures

F.1 Kendall correlations



(a) Text-level (b) System-level

Figure 4: Absolute Kendall correlations (%) between human evaluations (more readable view)

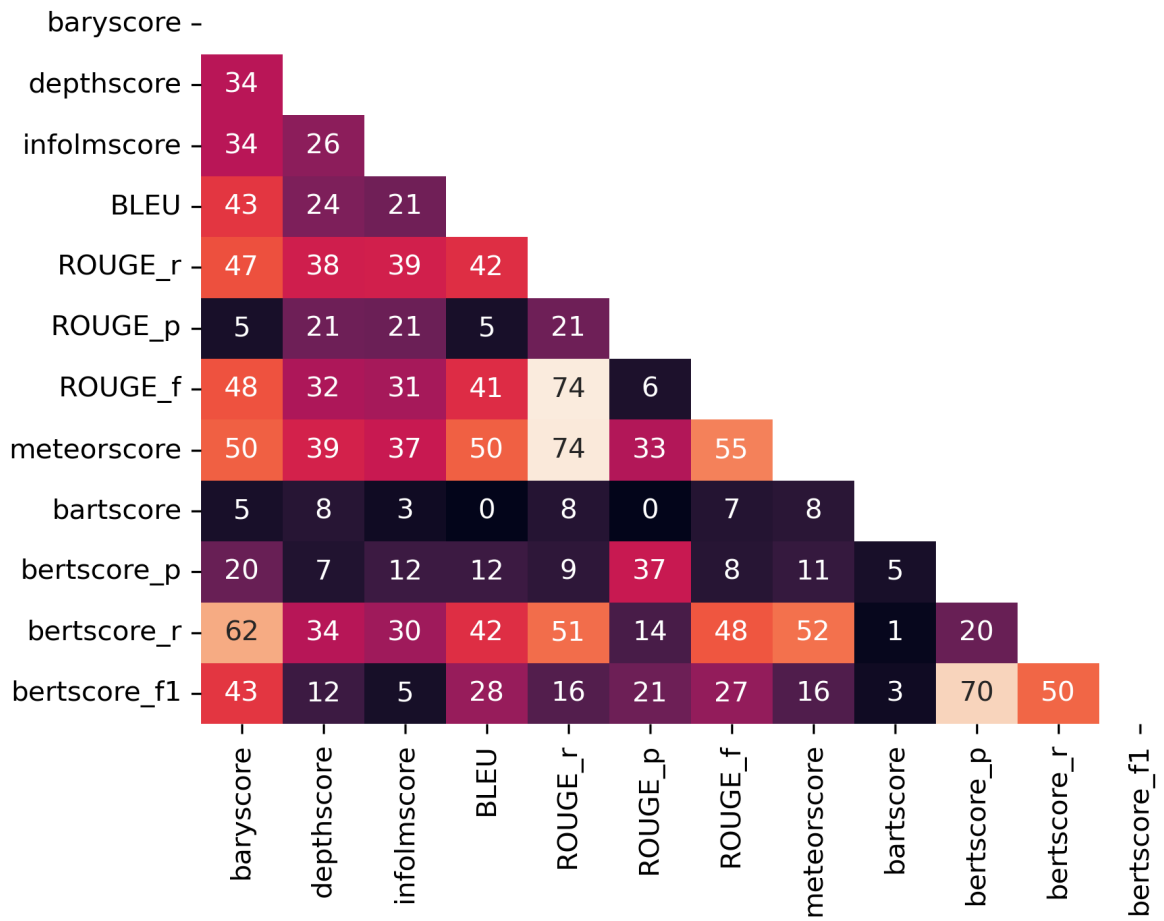


Figure 5: Absolute Kendall correlations (%) between automatic scores, text-level

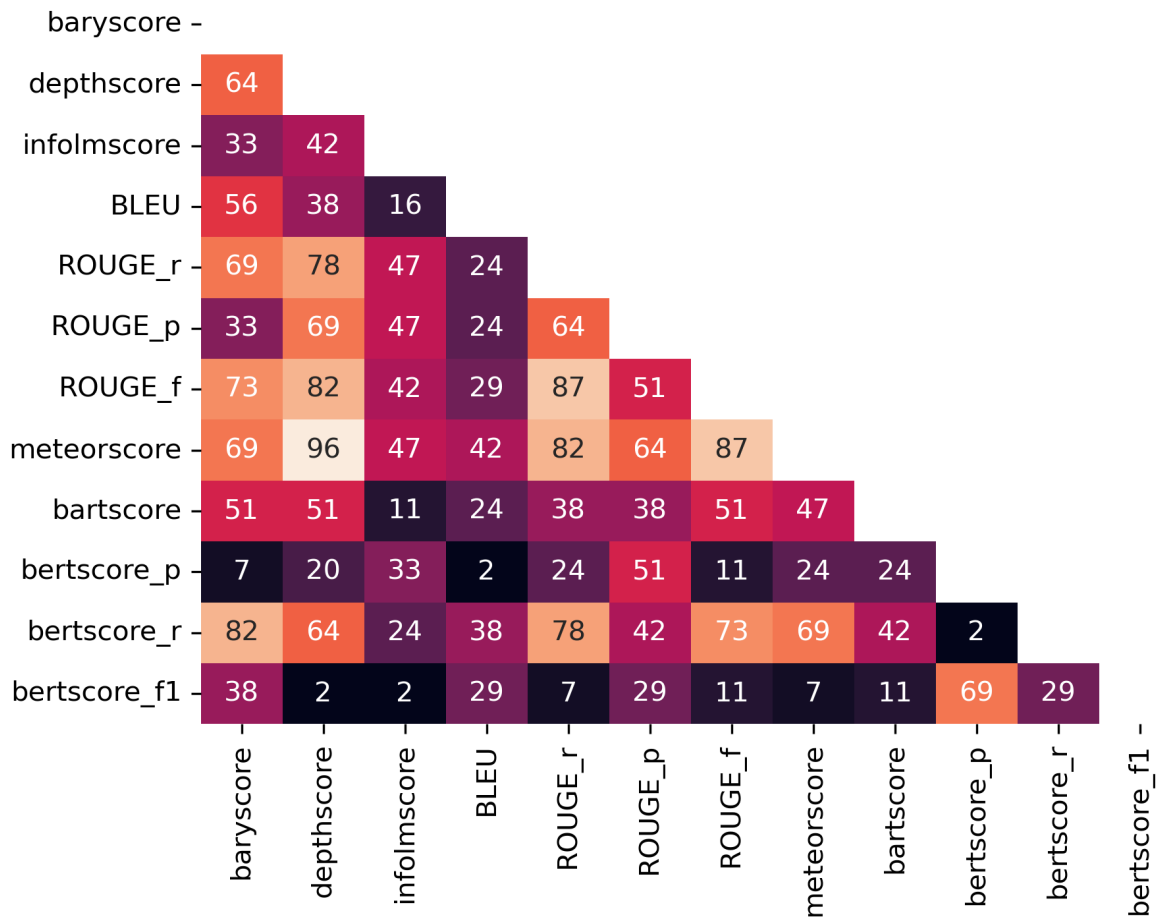
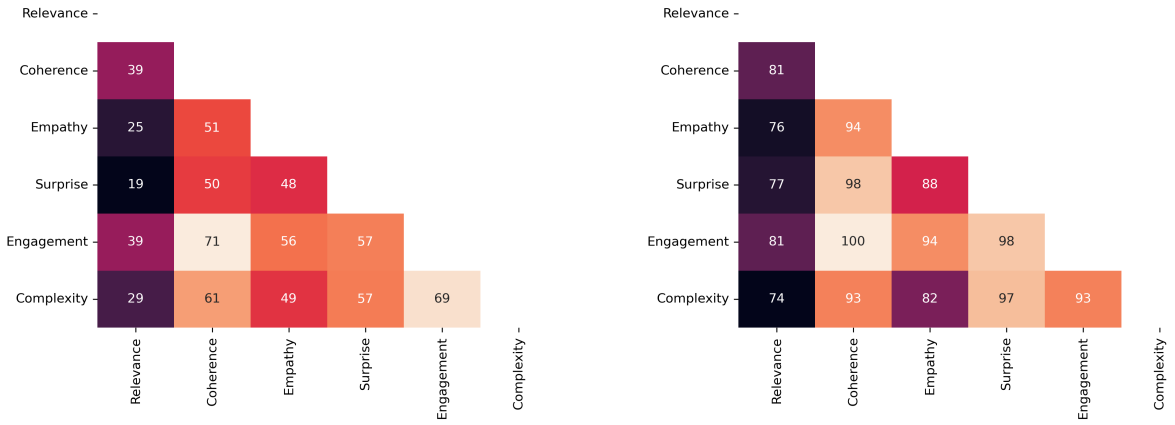


Figure 6: Absolute Kendall correlations (%) between automatic scores, system-level

F.2 Spearman correlations



(a) Text-level (b) System-level

Figure 7: Absolute Spearman correlations (%) between human evaluations

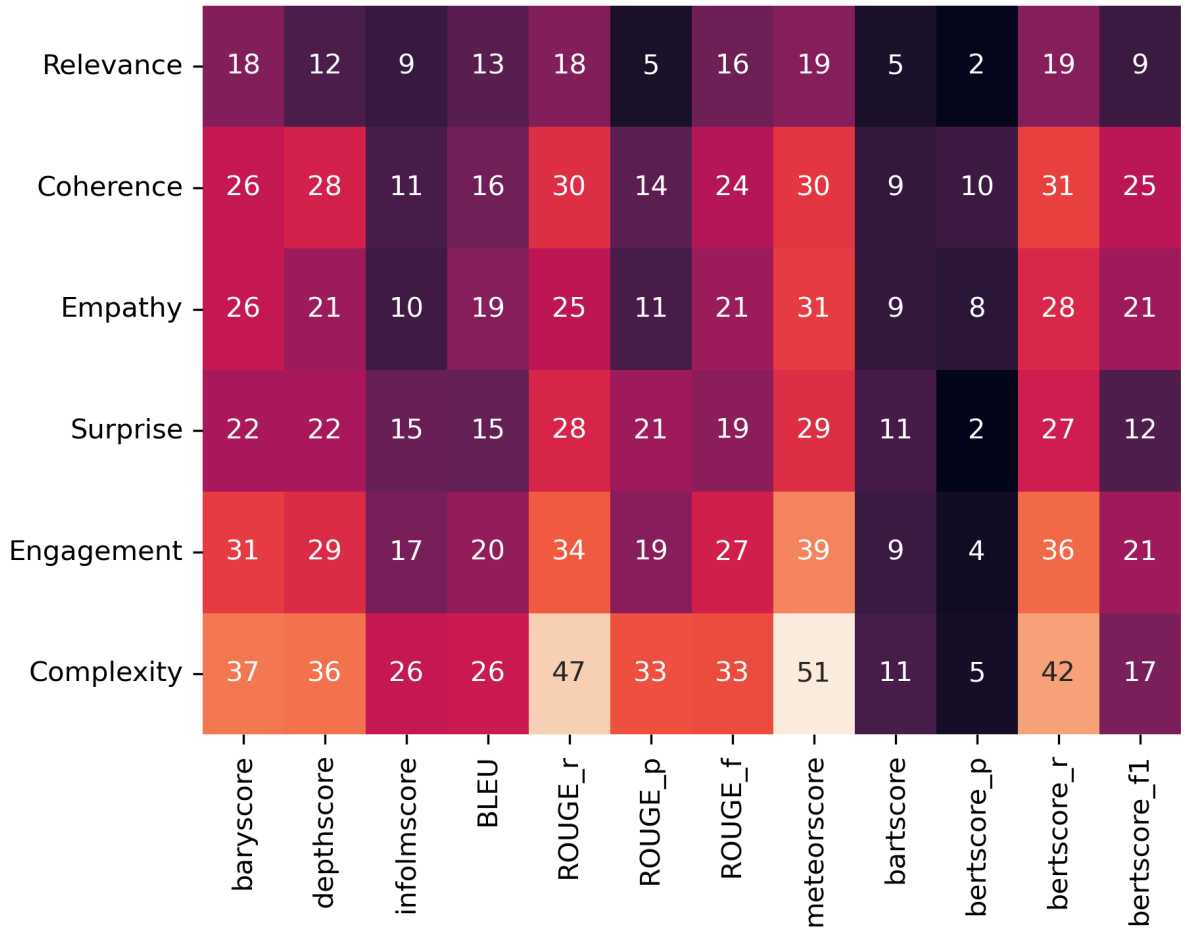


Figure 8: Absolute Spearman correlations (%) between human evaluations and automatic scores, text-level

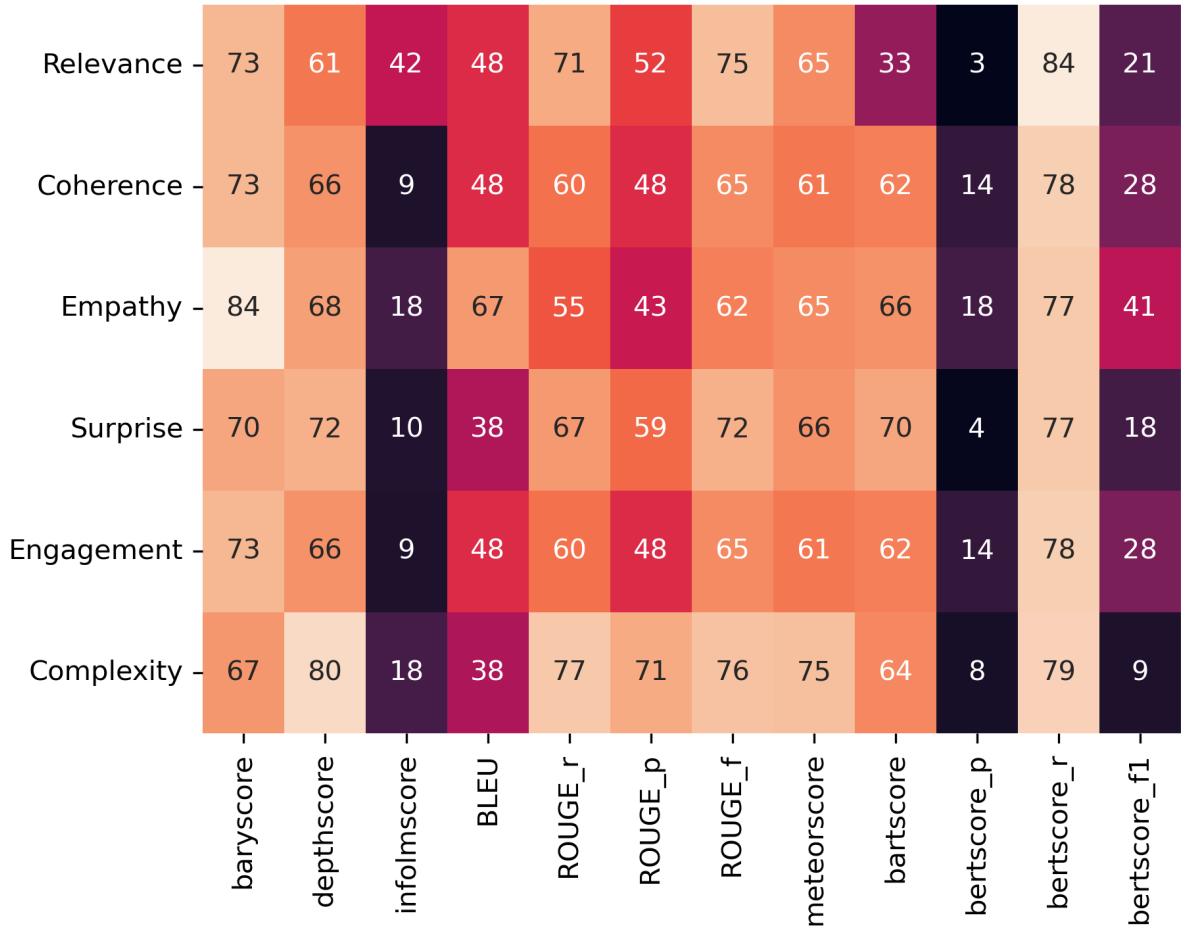


Figure 9: Absolute Spearman correlations (%) between human evaluations and automatic scores, system-level

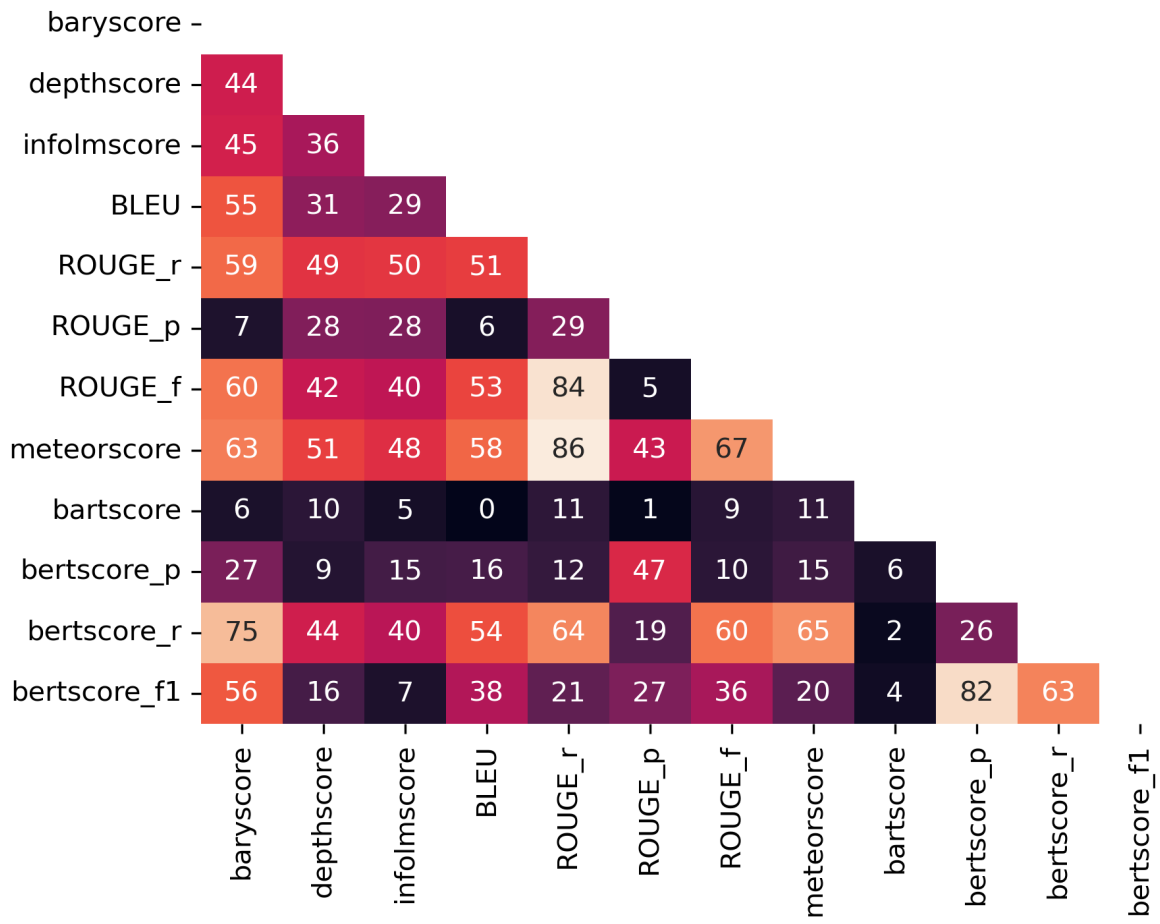


Figure 10: Absolute Spearman correlations (%) between automatic scores, text-level

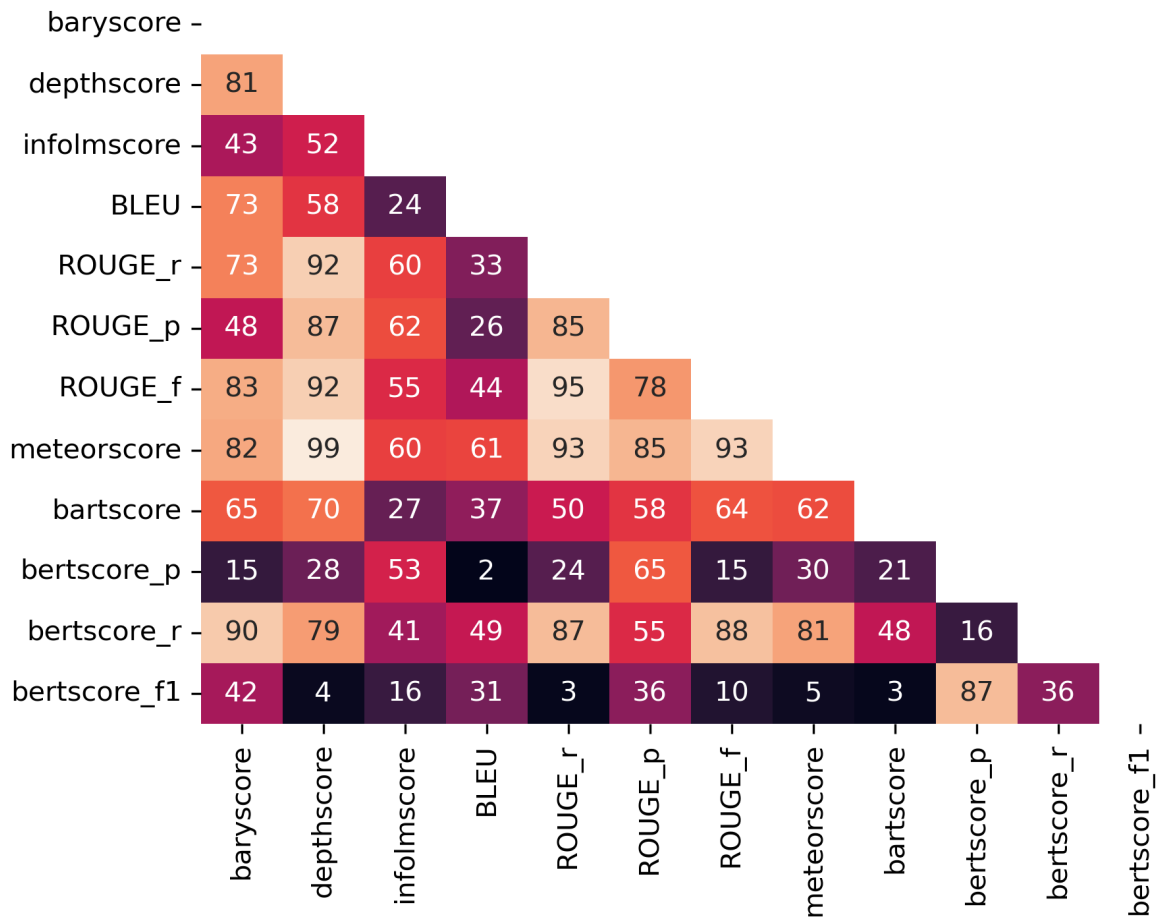
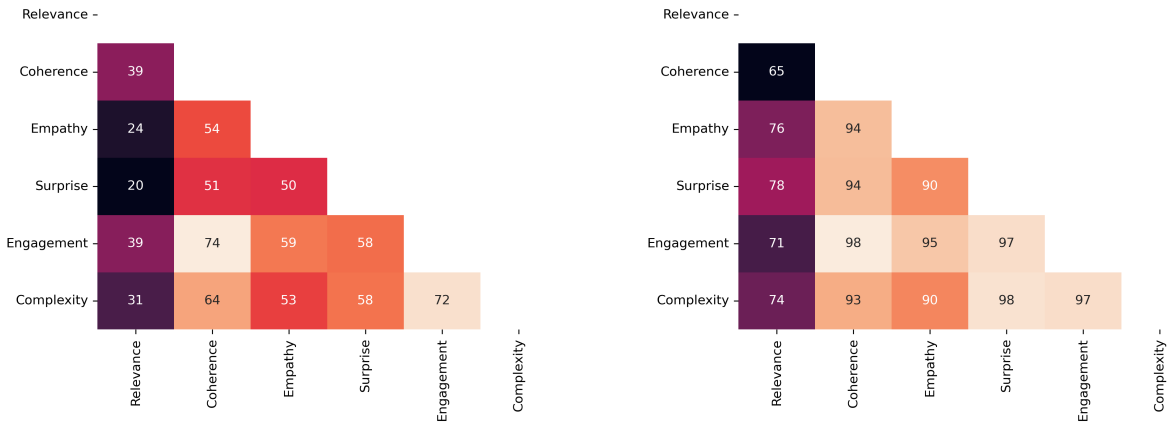


Figure 11: Absolute Sperman correlations (%) between automatic scores, system-level

F.3 Pearson correlations



(a) Text-level (b) System-level

Figure 12: Absolute Pearson correlations (%) between human evaluations

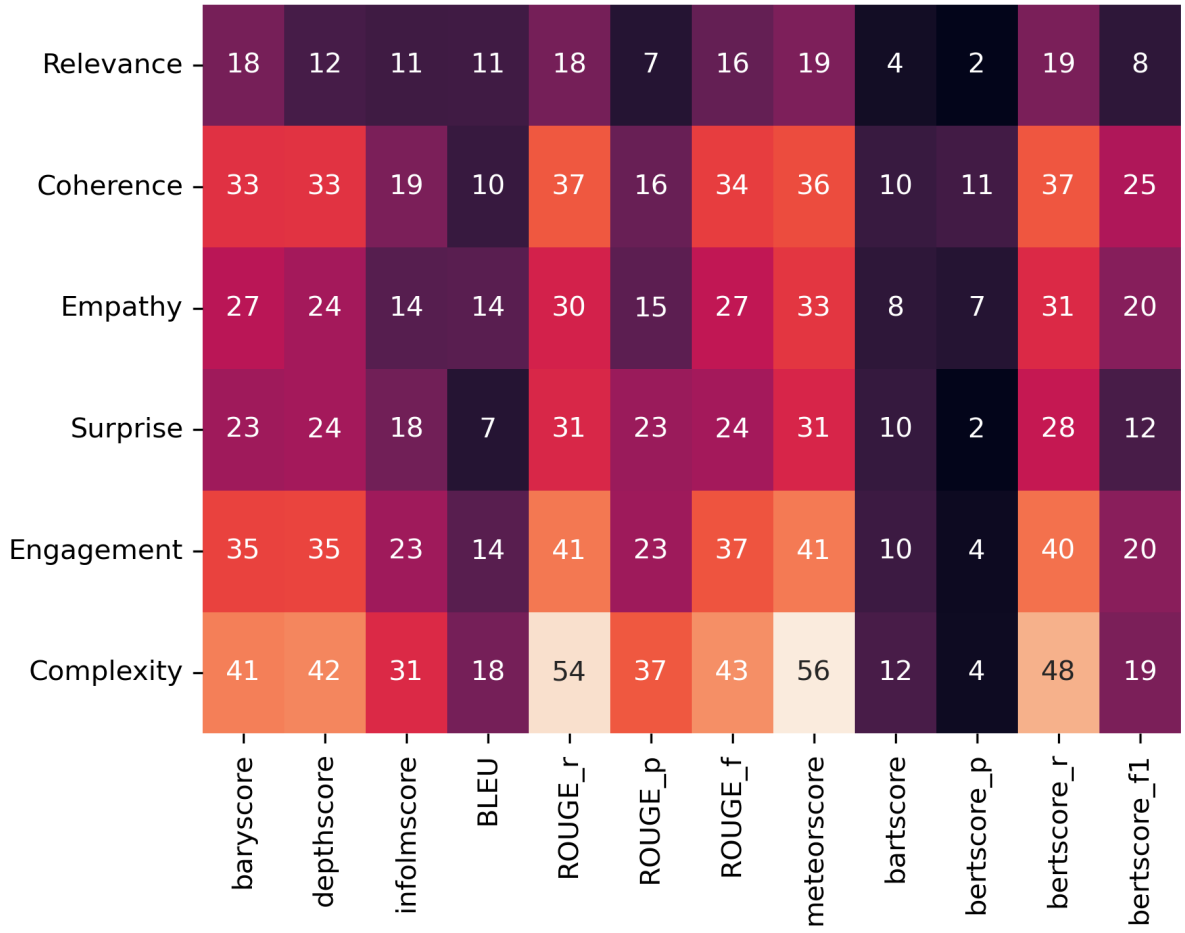


Figure 13: Absolute Pearson correlations (%) between human evaluations and automatic scores, text-level

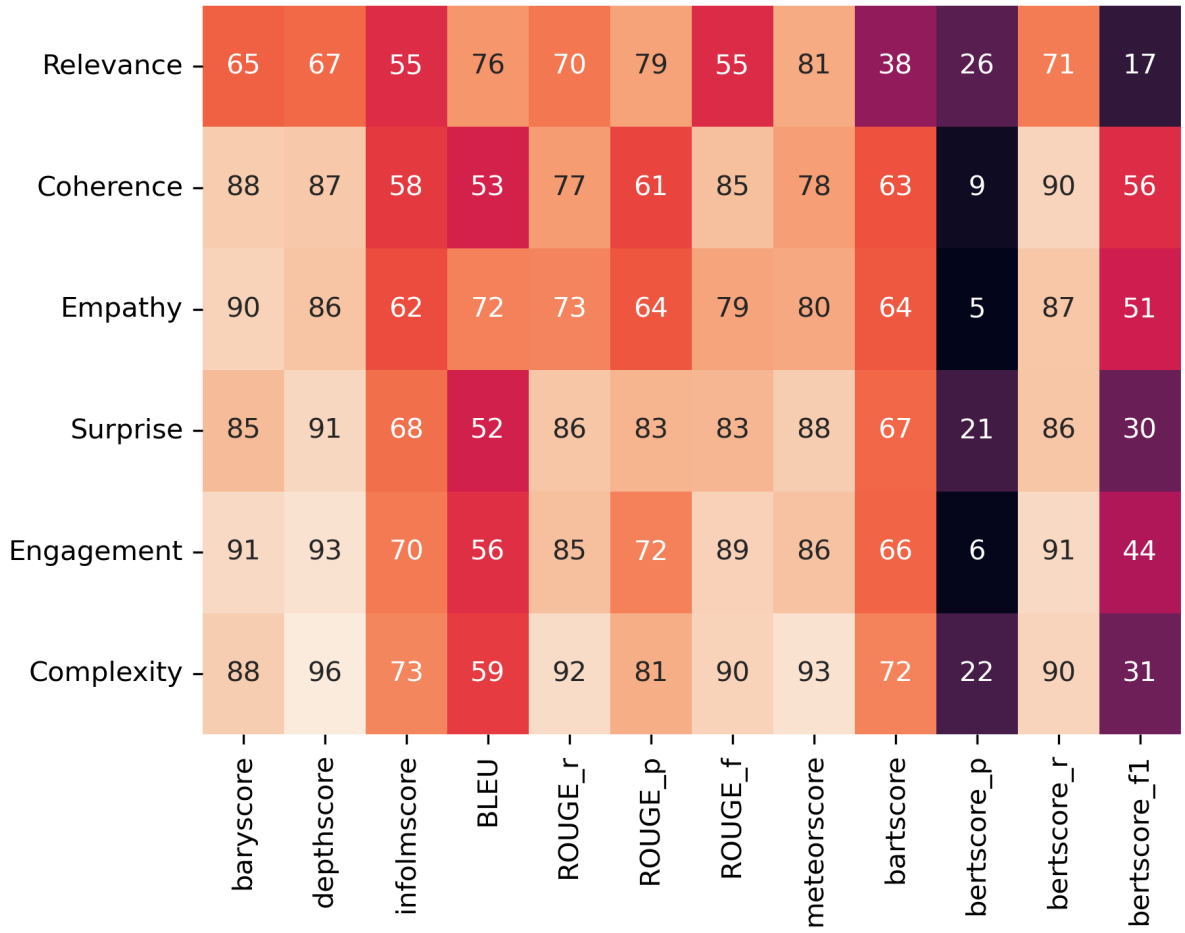


Figure 14: Absolute Pearson correlations (%) between human evaluations and automatic scores, system-level

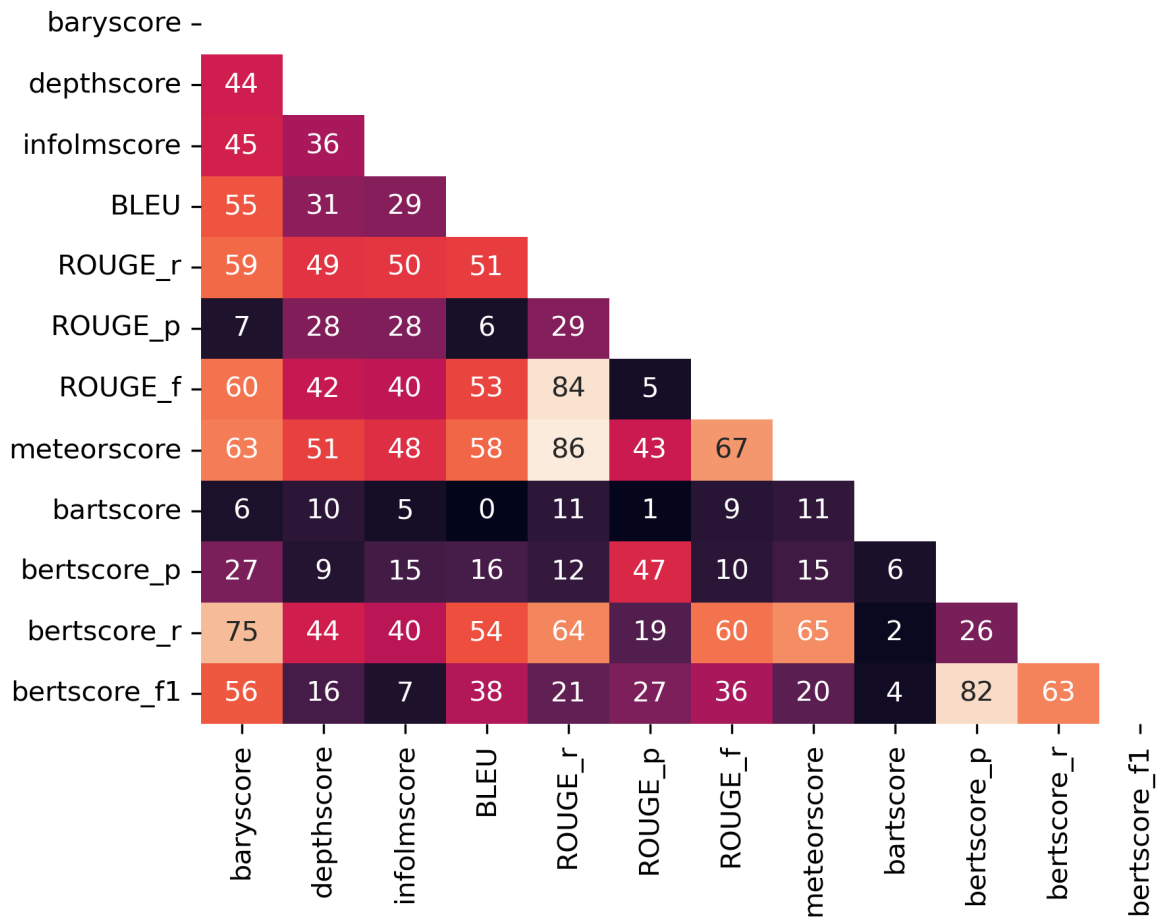


Figure 15: Absolute Pearson correlations (%) between automatic scores, text-level

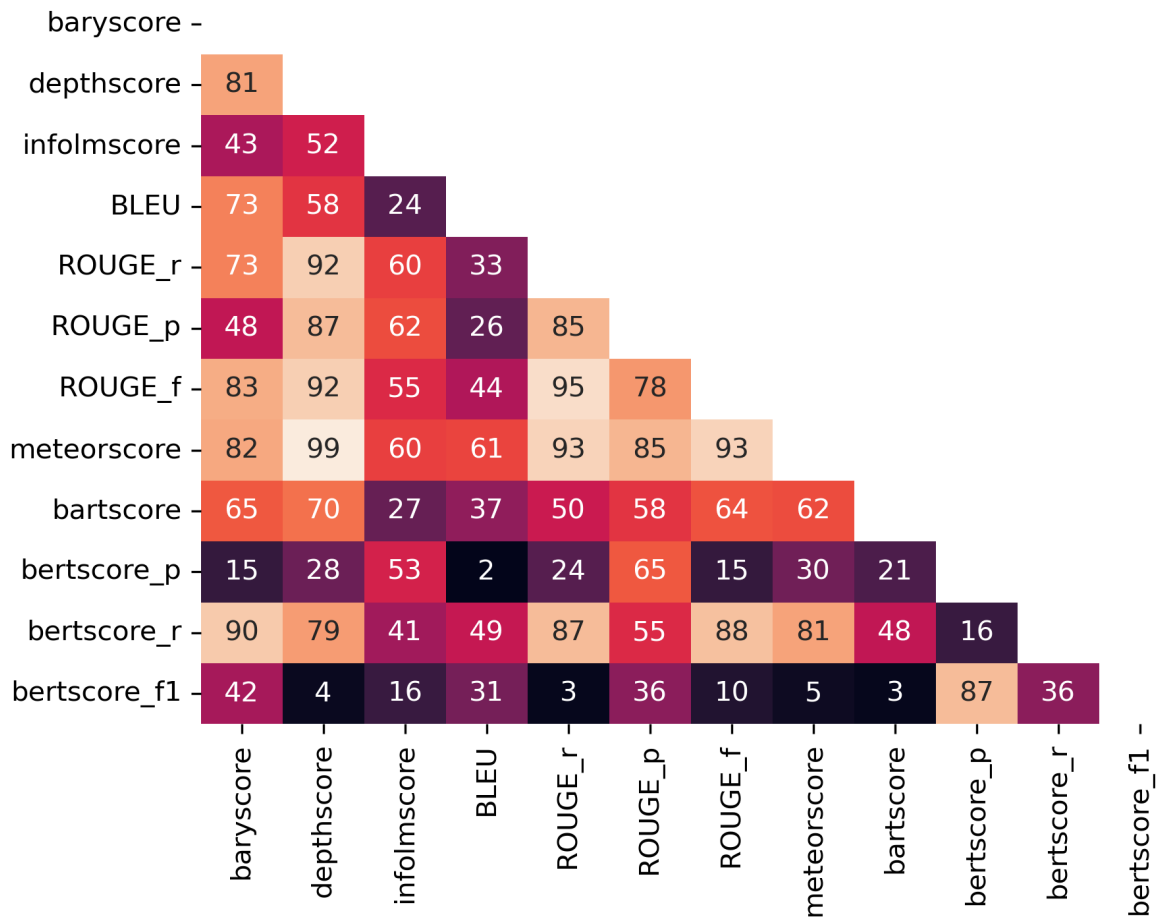


Figure 16: Absolute Pearson correlations (%) between automatic scores, system-level