Paper summaries

October 20, 2020

Chapter 1

On the Automatic Generation of Medical Imaging Reports (Jing, Xie, and Xing, 2018)

1.1 Introduction

The reading and interpretation of medical images are usually conducted by specialized medical professionals. Report writing can be error-prone for inexperienced physicians, and time-consuming and tedious for experienced physicians. Several challenges need to be addressed:

1. A complete report consists of multiple heterogeneous sources of information
2. Localize image regions and attach the right description to them
3. Descriptions in reports are usually long, with multiple sentences

The proposed solutions are:

1. A **multi-task learning framework** for simultaneous prediction of tags and text generation
2. A **Co-attention mechanism:** simultaneous attention to images and predicted tags; explores synergistic effects of visual and semantic information
3. A **Hierarchical LSTM:** Leverages compositional nature of reports: first generates high-level topics, then fine-grained descriptions from each one

1.2 Methods and Architecture

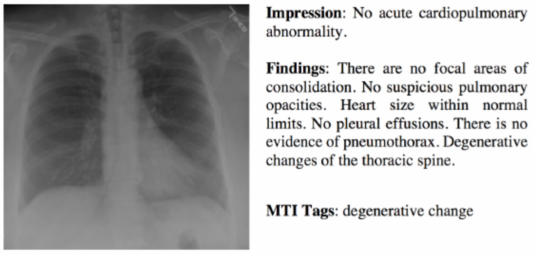
An image is divided into regions, and a CNN encoder is used to learn visual features for these patches. These features are fed into a *multi-label classifier*, from which tags are predicted. These tags are transformed into *semantic feature vectors* by a custom embedding. Both visual and semantic features are fed into the co-attention module, which produces a combined *context vector,* which **simultaneously captures the visual and semantic information of this image.**

The decoding and caption generation process is performed by the hierarchical LSTM, which leverages the compositional structure of a medical report (each sentence focusing on one specific topic). The *sentence LSTM,* using the context vector, first generates a sequence of high-level topic vectors representing sentences. Each one is passed to the *word LSTM,* which then generates a sentence for each topic vector. The number of sentences or topic vectors to be generated is regulated by the *stop control.*

1.2.1 Tag prediction

This is treated as a multi-label classification task. Given an image I, visual features $\{\boldsymbol{v}_n\}_{n=1}^{N} \in \mathbb{R}^D$ are extracted from the CNN encoder, and fed to a *multi-label classification* (MLC) network, which then generates a probability distribution over the L tags

$$\boldsymbol{p}_{I,\text{pred}}\left(\boldsymbol{l}_i = 1 \mid \{\boldsymbol{v}_n\}_{n=1}^{N}\right) \propto \exp\left(\text{MLC}_i\left(\{\boldsymbol{v}_n\}_{n=1}^{N}\right)\right)$$



Impression: No acute cardiopulmonary abnormality.

Findings: There are no focal areas of consolidation. No suspicious pulmonary opacities. Heart size within normal limits. No pleural effusions. There is no evidence of pneumothorax. Degenerative changes of the thoracic spine.

MTI Tags: degenerative change

Figure 1: An exemplar chest x-ray report. In the *impression* section, the radiologist provides a diagnosis. The *findings* section lists the radiology observations regarding each area of the body examined in the imaging study. The *tags* section lists the keywords which represent the critical information in the findings. These keywords are identified using the Medical Text Indexer (MTI).

Figure 1.1: Sample report from IU X-ray

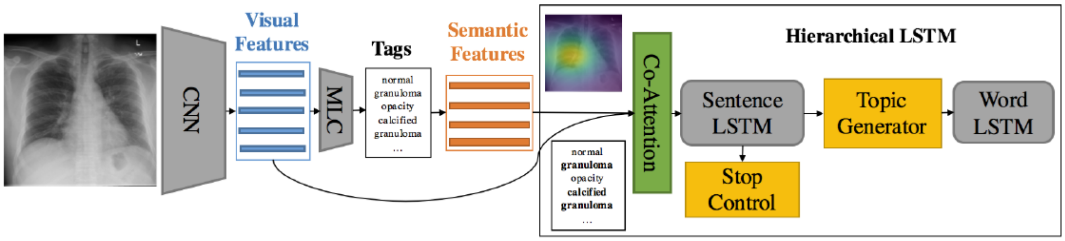


Figure 2: Illustration of the proposed model. MLC denotes a *multi-label classification* network. Semantic features are the word embeddings of the predicted tags. The boldfaced tags "calcified granuloma" and "granuloma" are attended by the co-attention network.

Figure 1.2: Architecture

where $\boldsymbol{l} \in \mathbb{R}^L$ is a binary tag vector, each component representing the presence or absence of the corresponding tag. Finally, the embeddings of the M most likely tags $\{\boldsymbol{a}_m\}_{m=1}^M$ are used as **semantic features.**

1.2.2 Co-Attention

Visual attention does not provide sufficient high-level semantic information, which the tags can always provide. A co-attention mechanism can simultaneously attend to visual and semantic modalities.

In the sentence LSTM at time step s, the joint context vector $\textbf{ctx}^{(s)} \in \mathbb{R}^C$ is generated by a co-attention network $f_{\text{co-att}}\left(\{\boldsymbol{v}_n\}_{n=1}^N, \{\boldsymbol{a}_m\}_{m=1}^M, \boldsymbol{h}_{\text{sent}}^{(s-1)}\right)$, with $\boldsymbol{h}_{\text{sent}}^{(s-1)} \in \mathbb{R}^H$ being the previous hidden state. The co-attention network $f_{\text{co-att}}$ uses a single feedforward layer to compute separate soft visual and semantic attentions

$$\alpha_{\mathbf{v},n} \propto \exp\left(\mathbf{W}_{\mathbf{v}_{\text{att}}}\mathbf{v}_n + \mathbf{W}_{\mathbf{v},\mathbf{h}}\mathbf{h}_{\text{sent}}^{(s-1)}\right)$$
$$\alpha_{\mathbf{a},m} \propto \exp\left(\mathbf{W}_{\mathbf{a}_{\text{att}}}\mathbf{a}_m + \mathbf{W}_{\mathbf{a},\mathbf{h}}\mathbf{h}_{\text{sent}}^{(s-1)}\right)$$

The visual and semantic context vectors

$$\mathbf{v}_{\text{att}}^{(s)} = \sum_{n=1}^{N} \alpha_{\mathbf{v},n}\mathbf{v}_n$$
$$\mathbf{a}_{\text{att}}^{(s)} = \sum_{m=1}^{M} \alpha_{\mathbf{a},m}\mathbf{a}_m$$

These context vector may be combined by concatenation followed by a fully connected layer:

$$\textbf{ctx}^{(s)} = \mathbf{W}_{\text{fc}}\left[\mathbf{v}_{\text{att}}^{(s)}; \mathbf{a}_{\text{att}}^{(s)}\right]$$

1.2.3 Sentence LSTM

A single LSTM layer whose input is the joint context vector $\textbf{ctx}^{(s)}$ and generates a topic vector $\mathbf{t} \in \mathbb{R}^K$as long as the stop control allows it to.

Topic generator Deep output layer (LSTM + multi-layer feedforward)

$$\mathbf{t}^{(s)} = \tanh\left(\mathbf{W}_{\mathbf{t},\mathbf{h}}\mathbf{h}_{\text{sent}}^{(s)} + \mathbf{W}_{\mathbf{t},\textbf{ctx}}\textbf{ctx}^{(s)}\right)$$

Stop control Deep output layer for the continuation of the sentence LSTM. The layer takes the previous and current hidden states and produces a distribution over $\{\text{STOP} = 1, \text{CONTINUE} = 0\}$, $p_{\text{stop}}^{(s)}$

$$p\left(\text{STOP} \mid \mathbf{h}_{\text{sent}}^{(s-1)}, \mathbf{h}_{\text{sent}}^{(s)}\right) \propto \exp\left\{\mathbf{W}_{\text{stop}} \tanh\left(\mathbf{W}_{\text{stop},s-1}\mathbf{h}_{\text{sent}}^{(s-1)} + \mathbf{W}_{\text{stop},s-1}\mathbf{h}_{\text{sent}}^{(s)}\right)\right\}$$

This stopping probability is then compared with a predefined threshold.

1.2.4 Word LSTM

For each topic vector, the words in the sentence are generated by the *word LSTM*. Its first and second inputs are the topic vector $\mathbf{t}$ and a START token, then followed by the rest of the words (Krause et al., 2016). Each hidden state $\mathbf{h}_{\text{word}} \in \mathbb{R}^H$ is directly used to predict the distribution over words:

$$p\left(\text{word} \mid \mathbf{h}_{\text{word}}\right) \propto \exp\left(\mathbf{W}_{\text{out}}\mathbf{h}_{\text{word}}\right)$$

1.2.5 Parameter learning

Each training example is a tuple $(I, \mathbf{l}, \mathbf{w})$, with $\mathbf{w}$ being the paragraph, with S sentences, each with T_S words (ground truth).

1. The MLC predicts the tag distribution $\mathbf{p}_{I,\text{pred}}$. Its ground truth may be computed by $\mathbf{p}_I = \mathbf{l} / \left\lVert \mathbf{l} \right\rVert_1$. Thus, the training loss for this task is the cross-entropy between both distributions ℓ_{tag}.

2. The sentence LSTM is unrolled for S steps, producing that number of topic vectors $\mathbf{t}^{(s)}$ and stop distributions $p_{\text{stop}}^{(s)}$. For each sentence, this stop probability is compared with the indicator $I\left\{s = S\right\}$, which evaluates to 0 [CONTINUE], until $s = S$, when it evaluates to 1 [STOP] (that is, a kronecker delta δ_{sS}). The cross entropy between them is the loss ℓ_{sent}.

3. The S topic vectors are fed to the word LSTM to generate $\mathbf{w}_{s,t}$ words. The training loss for each word is the cross entropy ℓ_{word} between the ground truth word $w_{s,t}$ and the predicted word distribution $p_{s,t}$.

Thus, the overall training loss is

$$\begin{aligned} \ell\left(I, \mathbf{l}, \mathbf{w}\right) = \lambda_{\text{tag}} \ell_{\text{tag}} \\ + \lambda_{\text{sent}} \sum_{s=1}^{S} \ell_{\text{sent}} \left(p_{\text{stop}}^{(s)}, I\left\{s = S\right\}\right) \\ + \lambda_{\text{word}} \sum_{s=1}^{S} \sum_{t=1}^{T_S} \ell_{\text{word}}\left(p_{s,t}, w_{s,t}\right) \end{aligned}$$

Furthermore, and attention regularization loss (Xu et al., 2015) for both visual $\boldsymbol{\alpha} \in \mathbb{R}^{N \times S}$ and semantic $\boldsymbol{\beta} \in \mathbb{R}^{M \times S}$ attention coefficients. This regularization encourages the model to pay equal attention over different image regions and tags.

$$\ell_{\text{reg}} = \lambda_{\text{reg}} \left[\sum_{n=1}^{N} \left(1 - \sum_{s=1}^{S} \alpha_{n,s}\right)^2 + \sum_{m=1}^{M} \left(1 - \sum_{s=1}^{S} \beta_{m,s}\right)^2 \right]$$

Chapter 2

Evaluation of text generation: A survey (Celikyilmaz, Clark, and Gao, 2020)

Surveys evaluation methods for Natural Language Generation (NLG) that may be classified as

- Human-centric evaluation
- Automatic metrics
- Machine-learned evaluation

2.1 Introduction

NLG evaluation is challenging mainly because many NLG tasks are open-ended (*i.e.* multiple valid answers for a task), so human evaluation remains the gold standard.

NLG techs range from template-based systems to machine learned systems that have a complex understanding of human grammar. There has been a paradigm shift from earlier template-based models using statistical methods and expert knowledge to unsupervised learning of representations from large textual corpora by using DNN models. This shift crosses through RNNs, (LSTM, GRU), word2vec, GloVe, and sequence-to-sequence by encoder-decoder architecture. These latter models' weakness of capturing long-range dependencies motivates the development of *attention networks* and *pointer networks.* The transformer architecture combines both encoder-decoder with a self-attention mechanism.

Neural models have been applied to many NLG tasks:

- summarization: documents, query-focused or generic, for news articles, meetings, etc
- machine translation
- dialog systems: goal-oriented or chit-chat
- question generation
- long text generation: story, news, poem
- data-to-text generation: *e.g.* table summarization
- **caption generation from non-text input:** tables, images or sequences of video frames

> *"Nevertheless, training a powerful language model relies on evaluation metrics that can measure the model quality from different perspectives. For instance, it is imperative to build evaluation methods that can determine whether a text is generated by a human or a machine to prevent any potential harm. Similarly, evaluating the generated text based on factual consistency has recently drawn attention in the NLG field. It is concerning that neural language models can generate open-ended texts that are fluent but not grounded in real-world knowledge or facts, such as fake news. The situation is particularly alarming if the generated reports or news are related to the well-being of humankind, such as summaries of health reports."*

2.1.1 Outline

- **Human-centric Evaluation:** *humans as judges.* Task-specific.
- **Untrained Automatic Metrics:** Most commonly used, compared model outputs to references with metrics based on string overlap, content overlap, string distance or lexical diversity.
- **Machine-learned Metrics:** models that can be viewed as digital judges that simulate human judges.

2.2 Human-Centric Evaluation Methods

The ultimate goal of NLG is to generate text that is valuable to people.
Human evaluations pose several challenges: expensive and time-consuming to run, especially for tasks that require domain expertise. There is also a lack of consistency in how these evaluations are run, which prevents reproducibility and comparing across systems.

2.2.1 Intrinsic Evaluation

An *intrinsic evaluation* asks people to evaluate the quality of generated text, either overall or along some specific dimension (e.g., fluency, coherence, correctness, etc.). This may be done by showing each text individually and asking for an assessment by binary (good/bad) or Likert/sliding scale. However, these judgments can be inconsistent and comparing these results is not straightforward.

Another approach to directly compare to baselines model variants, or human generated texts: evaluations can be performed by selecting or ranking generated texts. This captures models' relative quality but not absolute. This can be solved by asking judges to indicate how much better the chosen text is over alternatives. Comparison-based approaches can become prohibitively expensive (by number of matchups), though this may be somewhat mitigated.

Some dimensions of human evaluation include:

- *Adequacy: "how much of the meaning expressed in the gold-standard translation or source is also expressed in the target translation"*
- *Fluency:* quality of text without taking the source into account, accounting for grammar, spelling, choice of words, style, etc.
- *Factuality:* important in tasks that require the generated text to accurately reflect facts described in the context (many neural NLG models "hallucinate" information)
- Misc: *commonsense, logicality, coherence or consistency*

Other dimensions focus on *how* the text is being said, without context:

- *Grammaticality*
- *Style, formality* or *tone*
- *Typicality:* how often do you expect to see text that looks like this?
- *Redundancy*

2.2.2 Extrinsic evaluation

An *extrinsic evaluation* asks people to evaluate the system's performance on the task for which it was designed. They are the most meaningful for downstream tasks, but can be expensive and difficult to run.

2.2.3 The Evaluators

For many NLG tasks, no specific expertise is required of the evaluators (other than language). Many evaluators can be crowdsourced online (*e.g.* Amazon Mechanical Turk), though issues of quality control may be raised. Other cases require expert human annotators.

2.2.4 Inter-Evaluator Agreement

Evaluating natural language will always include some degree of subjectivity. The rate of evaluator disagreement is also useful to measure, since high agreement can signal the task is well-defined and differences in generated text are noticeable, while low agreement can mean the opposite.

2.2.4.1 Percent agreement

Flat percent of cases in which the evaluators agree: X is the set of generated texts, and $a_i = 1$ indicates agreement

$$P_a = \frac{1}{|X|}\sum_{i=0}^{|X|} a_i \tag{2.1}$$

This may be a common metric, but does not take into account chance agreement, especially with small number of scores (hypothesis testing?)

2.2.4.2 Cohen's κ

This measure can capture agreements that may happen by chance: we now consider such probabilityP_c. For **two** evaluators are scoring X with a score from a set S, the probability of all scoring a text the same (assuming independence):

$$P_c = \sum_{s\in S} P\left(s \mid e_1\right) P\left(s \mid e_2\right)$$

Each $P\left(s \mid e_i\right)$ is estimated by the frequency with which evaluator e_i assigned each score s. Cohen's κ is defined as

$$\kappa = \frac{P_a - P_c}{1 - P_c}$$

2.2.4.3 Fleiss' κ

Generalizes Cohen's κ for more than two raters, by considering agreement across all pairs of evaluators. We now define the agreement a_i for each text x_i as

$$a_i = \frac{\sum_{s\in S} \text{\# of evaluator pairs who score } x_i \text{as } s}{\text{total \# of evaluator pairs}}$$

The agreement probability P_a can thus be computed by 2.1.

The chance agreement probability is estimated by aggregating the frequency of scores across all annotators, with the assumption that each judge is equally likely to draw from this distribution r_s. Thus, the chance of two annotators assigning score s by chance is r_s^2, and the overall probability is

$$P_c = \sum_{s\in S} r_s^2$$

Fleiss' κ is once again defined as

$$\kappa = \frac{P_a - P_c}{1 - P_c}$$

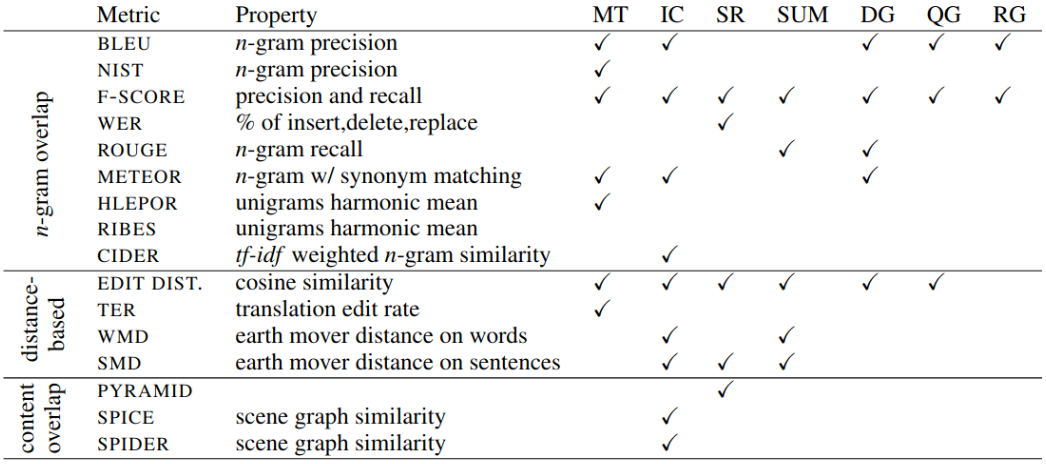
2.2.4.4 Krippendorff's α

This is technically a measure of *disagreement*, allowing different levels of disagreement to be taken to account.

The overall probability of disagreement across all possible score pairs (s_m, s_n), each weighted by $w_{m,n}$ is computed by

$$P_d = \sum_{m=0}^{|S|}\sum_{n=0}^{|S|} w_{m,n} \sum_{i=0}^{|X|} \frac{\text{\# of evaluator pairs that assign } x_i \text{as } (s_m, s_n)}{\text{total \# of evaluator pairs}}$$

(noting that $w_{m,n} = 0$ for score agreement).



	Metric	Property	MT	IC	SR	SUM	DG	QG	RG
n-gram overlap	BLEU	*n*-gram precision	✓	✓			✓	✓	✓
	NIST	*n*-gram precision	✓						
	F-SCORE	precision and recall	✓	✓	✓	✓	✓	✓	✓
	WER	% of insert,delete,replace			✓				
	ROUGE	*n*-gram recall				✓	✓		
	METEOR	*n*-gram w/ synonym matching	✓	✓			✓		
	HLEPOR	unigrams harmonic mean	✓						
	RIBES	unigrams harmonic mean							
	CIDER	*tf-idf* weighted *n*-gram similarity		✓					
distance-based	EDIT DIST.	cosine similarity	✓	✓	✓	✓	✓	✓	
	TER	translation edit rate	✓						
	WMD	earth mover distance on words		✓		✓			
	SMD	earth mover distance on sentences		✓	✓	✓			
content overlap	PYRAMID				✓				
	SPICE	scene graph similarity		✓					
	SPIDER	scene graph similarity		✓					

Table 3.1: Untrained *automatic* and *retrieval-based* metrics based on string match, string distance, or context overlap. The acronyms for some of the NLP sub-research fields that each metric is commonly used to evaluate text generation are: **MT**: Machine Translation, **QG**: Question Generation, **SUM**: Summarization, **RG**: Dialog Response Generation, **DG**: Document or Story Generation, Visual-Story Generation, etc., **IC**: Image Captioning.

Figure 2.1: Untrained metrics comparison

To calculate the expected disagreement, we also assume independence and estimate the probability of score assignment from the overall frequency. With the distribution of pairs $r_{m,n}$, the probability of disagreement by chance is

$$P_c = \sum_{m=0}^{|S|} \sum_{n=0}^{|S|} w_{m,n} r_{m,n}$$

Krippendorff's α is defined as

$$\alpha = 1 - \frac{P_d}{P_c}$$

2.3 Untrained Automatic Evaluation Metrics

As human-based evaluation is costly and time-consuming, automatic metrics are employed in developing and comparing new models against the state-of-the-art.

Untrained automatic metrics compute a score that indicates similarity between a generated text and a (human-written) reference. These metrics are fast and efficient.

2.3.1 n-gram Overlap Metrics for Content Selection

n-gram overlap metrics measure the degree of "matching" between generating and reference text.

2.3.1.1 F-Score F_1

$$F_1 = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where Precision is the fraction of n-grams in generated text that are present in reference text, and recall is the fraction of n-grams from the reference text that are present in the candidate text.

Character n-gram F-score seems some usage in machine translation, with high correlation with human judgments.

2.3.1.2 BLEU

Bilingual Evaluation Understudy is a weighted geometric mean of n-gram precision scores

$$\text{prec}_n = \frac{\sum_s \min\left[c\left(s,\hat{y}\right).c\left(s,y\right)\right]}{\sum_s c\left(s,\hat{y}\right)}$$

where $\hat{y}$ is the candidate sequence, y the reference, s an n-gram sequence of $\hat{y}$, and $c\left(s,\hat{y}\right)$ is the number of times s appears in $\hat{y}$.

Thus, BLEU is defined as

$$\text{BLEU} = \mathcal{BP} \times \exp\left(\sum_{n=1}^{N} w_n \log \text{prec}_n\right)$$

where $\mathcal{BP}$ is a *brevity penalty* that penalizes sequences that are too short

$$\mathcal{BP} = \begin{cases} 1 & c > r \\ e^{(1-r/c)} & c \leq r \end{cases}$$

where c is the candidate text length and r is the effective corpus reference length. N is the largest n-gram length (*i.e.* BLEU-N), and the weights w_n are usually taken as $1/N$.

While BLEU has some advantages for machine translation, it doesn't correlate well with human judgments for other tasks, such as image captioning and dialog response generation. It has been reported that "*there's not good evidence to support that* BLEU *is the best metric for evaluating NLG systems other than machine translation.* Even generated samples with perfect BLEU with the corpus and are grammatically correct, lacked semantic and/or global coherence.

2.3.1.3 NIST

Similar to BLEU for evaluating quality of text, but weights heavily less frequent n-grams, as co-occurrences of these are more informative than common n-grams.

2.3.1.4 ROUGE

Recall-Oriented Understudy for Gisting Evaluation evaluate automatic summarization of long texts with multiple sentences or paragraphs. Has also been adopted for machine translation, image captioning, etc.

Includes a number of variants, ROUGE-N also measures n-gram overlap, and can be defined as

$$\text{ROUGE-N} = \frac{\sum_r \sum_n \text{match}\left(\text{gram}_{n.r}\right)}{\sum_r \sum_n \text{count}\left(\text{gram}_n\right)}$$

where $\sum_n$ sums over all n–grams of length n, and $\sum_r$ repeats the sum for each reference summary, if there is more than one.

Some commonly used varieties:

- ROUGE-1,2,3,4: for corresponding n-grams
- ROUGE-L: measures the longest matching sequence of words using longest common subsequence.

Compared to BLEU, ROUGE focuses on recall rather than precision and is more interpretable. It evaluates the adequacy of generated text by counting how many n-grams in the text match the ones in the reference. This is considered a bottleneck, as it doesn't provide information about the narrative flow, grammar or topical flow; or evaluate factual correctness.

2.3.1.5 METEOR

Metric for Evaluation of Translation with Explicit ORdering is designed to address issues in BLEU and has been used for image captioning, question generation and summarization. It is based on the harmonic mean of unigram precision and recall, with recall being weighted higher. It yields high correlation with human judgments.

Let $c(u)$ be the number of unigrams matching between the hypothesis and reference text, and $c_h\left(u\right), c_r\left(u\right)$ be the total number of unigrams in the hypothesis and reference text respectively. Thus, Recall $= c\left(u\right)/c_r\left(u\right)$ and Precision $=$ $c\left(u\right)/c_h\left(u\right)$. Then

$$F_\alpha = \frac{\text{Precision} \cdot \text{Recall}}{\alpha\text{Precision} + (1-\alpha)\,\text{Recall}}$$

The alignment between hypothesis and reference is

$$\text{METEOR} = F_\alpha \times (1 - \text{penalty})$$

where the *fragmentation penalty* determines the extent to which the matched unigrams are well-ordered

$$\text{penalty} = \gamma\,\text{frag}^\beta$$

Bibliography

Celikyilmaz, Asli, Elizabeth Clark, and Jianfeng Gao (2020). *Evaluation of Text Generation: A Survey*. arXiv: 2006.14799 [cs.CL].

Jing, Baoyu, Pengtao Xie, and Eric Xing (July 2018). “On the Automatic Generation of Medical Imaging Reports”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2577–2586. DOI: 10.18653/v1/P18-1240. URL: https://www.aclweb.org/anthology/P18-1240.

Krause, Jonathan, Justin Johnson, Ranjay Krishna, and Li Fei-Fei (2016). *A Hierarchical Approach for Generating Descriptive Image Paragraphs*. arXiv: 1611.06607 [cs.CV].

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio (2015). *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. arXiv: 1502.03044 [cs.LG].