732A81/TDDE16 Text Mining (HT2024)

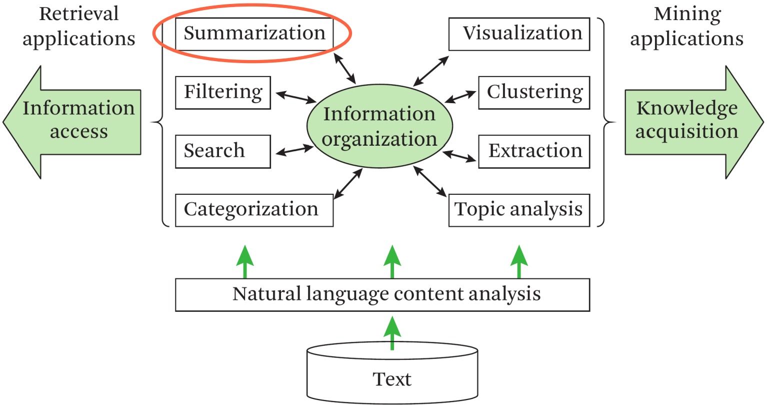
Text Summarization

Marcel Bollmann

Department of Computer and Information Science (IDA)

LINKÖPING UNIVERSITY

Reminder: Conceptual framework



Zhai and Massung (2016)

Outline

1. Introduction
   - Examples
   - Types of Summarization

2. Extractive Summarization
   - Sentence Transformers
   - Maximum Marginal Relevance (MMR)

3. Large Language Models
   - Causal Language Modelling
   - Instruction Fine-Tuning
   - Running LLMs

4. Abstractive Summarization

5. Evaluation
   - ROUGE
   - BERTScore

What is Text Summarization?

1

Text summarization

Definition

Text summarization is the task of **compressing** a relatively large amount of text data into a **more concise form** for easy digestion.

- Can help users **find relevant information** in large amounts of text.
- Challenges:
  1. **Identifying** the most important pieces of information.
  2. Producing text that is **coherent** and **meaningful**.

Zhai and Massung (2016)

Example: Summarizing scientific articles

SEMANTIC SCHOLAR

llama-2

Search

Llama 2: Open Foundation and Fine-Tuned Chat Models

Hugo Touvron · Louis Martin · +65 authors · Thomas Scialom · Computer Science · arXiv.org ·

18 July 2023

TLDR This work develops and releases Llama 2, a collection of pretrained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters, which may be a suitable substitute for closed-source models.Expand

1,030 · [PDF] · arXiv · Save · Cite

Financial News Analytics Using Fine-Tuned Llama 2 GPT Model

B. Pavlyshenko · Computer Science · arXiv.org · 24 August 2023

TLDR The obtained results show that the fine-tuned Llama 2 model can perform a multitask financial news analysis with a specified structure of response, part of response can be a structured text and another part of data can have JSON format for further processing.Expand

2 · [PDF] · arXiv · Save · Cite

Source: Semantic Scholar

Example: Summarizing opinions


goodreads Home My Books Browse ▾ Community ▾ Search books

VINTAGE MISHIMA
Spring Snow

Want to read
Buy on Amazon
Rate this book

Search review text
Filters
Displaying 1 - 10 of 1,548 reviews

Jim Fonseca
1,104 reviews
7,234 followers
Follow

★★★★★ February 27, 2019

Mishima (1925-1970) was a classic Japanese author. He was a fierce anti-communist who led a band of rebels trying to restore the Emperor. He committed ritual suicide when the plot failed. His best-known work is a tetralogy, The Sea of Fertility, of which this book is the first volume.

Class divisions and changing values in Japan due to western influence aree major themes. The main character is the son of a very wealthy family. How wealthy? They have 40 servants and the boy doesn't know all their names even though some of them who have worked there for years. The family lives in a multi-house compound which includes a western-style home in which they entertain and occasionally have western meals. The boy's own room is in a Japanese-style house but it's decorated with western furniture. His mother often dresses and wears her hair in western style. And yet, he has a friend at school whose family, he feels is, more 'western in outlook' than his, despite their family's lack of western trappings in furniture, food and dress.

The boy is very good-looking, dreamy, melancholy, lazy and ambition-less. He's bright but he figures he'll get into a university for rich kids because he's not wasting time studying for exams to get into an academically prestigious school. He's 18 when the story opens. The end of the Russo-Japanese War, 7 years ago, is a key backdrop to the story. So we know it's around 1912.

So the boy's family has money, but it's not one of Japan's traditional 28 noble families, like the one next door. So his father creates an alliance between those two families and the boy spends much time at the neighboring residence absorbing the noble ambiance. His father supplies the money; the other family supplies the prestige.

The plot revolves around a love story between this boy and the daughter of the neighboring household. They have known each other all their lives and she has loved him since they were children. But his feelings toward her are on-again, off-again; he mistreats her and pretends he doesn't care for her. Finally she gives up on him and becomes engaged to a son of a noble family, actually a member of the Emperor's household.

At this point (she's is 21; he's 19), and after the engagement has been approved by the Emperor himself, finally he decides he loves her and begins to pursue her. They begin a sexual relationship and she becomes pregnant. If word of any of this gets out, it would be the equivalent of a national scandal! When the boy's father learns what is going on, after spending his whole life ass-kissing the emperor and the nobles, to say he is apoplectic is putting it mildly. Never having lifted a hand to his son before, he beats him with a pool cue.

The difficult romance gives the author a chance to discuss the theme of 'the light of reason vs. the darkness of passions.' There's also quite a bit of discussion about Buddhism and reincarnation. But we know all this can only end in tragedy. His friend counsels him that he is throwing his life away almost as if he wants to commit suicide.

There is good writing, such as this passage that I liked: "On a warm spring day, a galloping horse was only too clearly a sweating animal of flesh and blood. But a horse racing through a snowstorm became one with the very elements; wrapped in the whirling blast of the north wind, the beast embodied the icy breath of winter."

It's a good story. I don't know if it entices me to read the whole tetralogy, but the second volume is the series, Runaway Horses, is equally highly rated as Spring Snow. (The other two are The temple of Dawn and The Decay of the Angel.) Probably the author's best-known work in English is not part of the tetralogy, it's The Sailor Who Fell from Grace with the Sea.

The review discusses Yukio Mishima's novel "Spring Snow," the first volume of the tetralogy "The Sea of Fertility." [...] **The review highlights the themes of reason vs. passion** and incorporates elements of Buddhism and reincarnation. **While the book is well-written and engaging, the reviewer is uncertain if it compels them to read the entire tetralogy** but mentions that the second volume, "Runaway Horses," is highly rated.

Source: Jim Fonseca on Goodreads and ChatGPT

Two approaches to summarization

- **Extractive summarization** extracts parts of the full document into a summary.
  - Parts = individual phrases or sentences
  - Challenges: coherence; cohesion; dependent on the style of the source material

- **Abstractive summarization** produces entirely new text that did not exist in the source document.
  - This comes closer to how humans produce summaries!
  - Requires more advanced methods to produce, e.g. LLMs

Extractive Summarization

2

Extractive summarization: Example

Original text

Orbital is a 2023 novel by English writer Samantha Harvey that incorporates elements of science fiction, literary fiction, and philosophical drama. It was published by Jonathan Cape in the UK and by Grove Atlantic in the US. It follows six fictional astronauts over 24 hours on the International Space Station, while including speculative interludes featuring an alien, a robot, and a prehistoric human. The novel was well received by critics. It won the 2024 Booker Prize and the Hawthornden Prize, and was nominated for...

Extractive summary

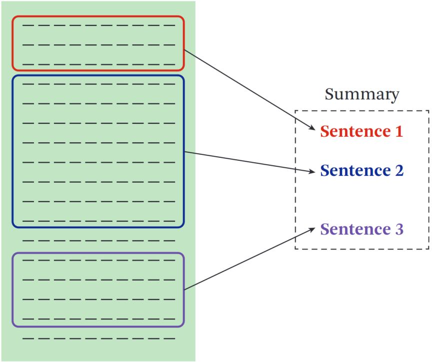
Orbital is a 2023 novel by English writer Samantha Harvey that incorporates elements of science fiction, literary fiction, and philosophical drama. The novel was well received by critics.

Source: Wikipedia

Extractive summaries with information retrieval

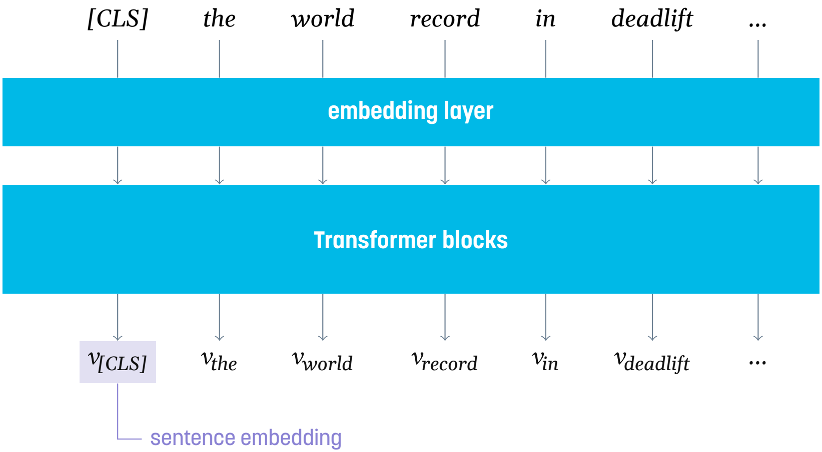
1. **Split the document** to be summarized into sections or passages.
   - *e.g.* paragraphs, or perform topic analysis

2. Compress the sentences in each passage into a smaller number of sentences that are **"relevant yet not redundant"**.
   - Can be framed as a ranking problem

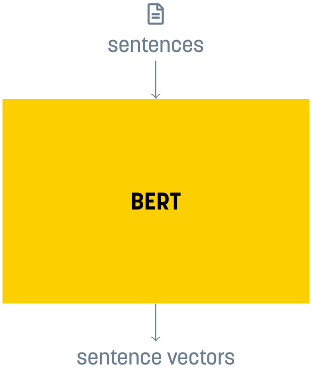


Zhai and Massung (2016), Fig. 16.2

Reminder: Embeddings from BERT models

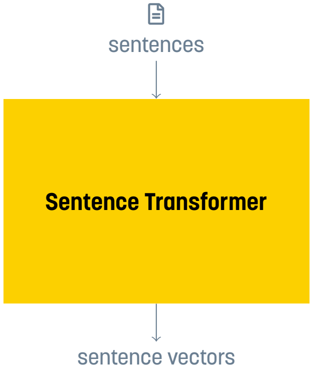


Reminder: Embeddings from BERT models



- We can feed a sentence into BERT and **extract embedding vectors** from it.
  - The special `[CLS]` token serves as an embedding for the entire sentence.

Sentence transformers



- **Sentence transformer** models are fine-tuned specifically for the purpose of extracting sentence embeddings.
  - *e.g.* using contrastive learning: “does this pair of sentences belong together or not?”

- The **Sentence Transformers** library provides an easy way to use these models.

Maximum marginal relevance

- We need a **ranking algorithm** that ranks sentences by their relevance.
  - For extractive summarization, we can then always pick the highest-ranked sentence.
- **Maximum marginal relevance (MMR)** is a classic (re-)ranking algorithm.
  - This algorithm *maximizes relevance* while *minimizing redundancy*.
- MMR compares sentences using a **similarity function**.
  - *e.g.* cosine similarity of sentence embeddings (or tf–idf vectors)

Maximum marginal relevance: Formula (I)

- The **next sentence** is picked using the following formula:

similarity function

$$s_i = \arg\max_{s \in R \setminus S} \left((1-\lambda) \cdot \mathrm{sim}(s, p) - \lambda \cdot \max_{s_j \in S} \mathrm{sim}(s, s_j) \right)$$

not yet selected sentences

profile vector that determines "relevance"

Maximum marginal relevance: Formula (II)

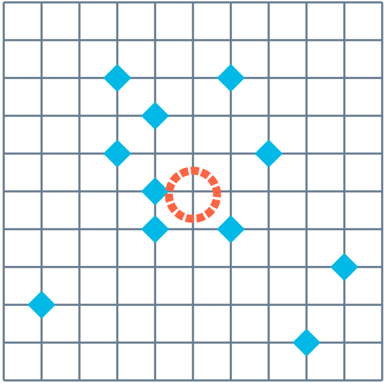
- MMR aims to **maximize relevance** while **minimizing redundancy**.

$$s_i = \underset{s \in R \setminus S}{\arg\max} \left((1 - \lambda) \cdot \mathrm{sim}(s, p) - \lambda \cdot \max_{s_j \in S} \mathrm{sim}(s, s_j) \right)$$

already selected sentences

- The **parameter** $\lambda \in [0, 1]$ weighs relevance against redundancy.

Maximum marginal relevance: Example

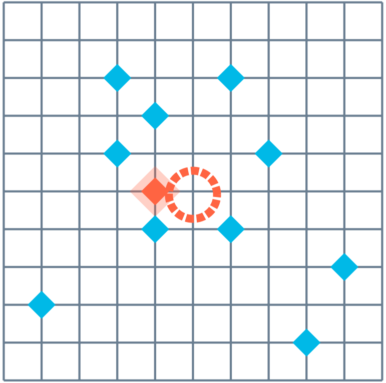


Let's assume each represents a sentence vector in our document.

For the **profile vector** p, a simple choice is to use the **centroid** of all sentence vectors in the document.

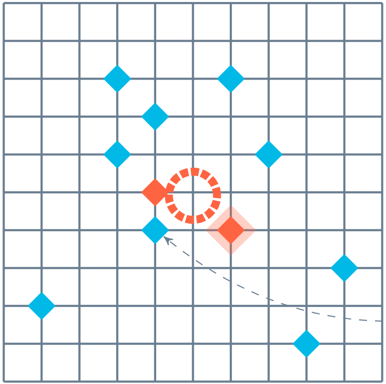
- Alternative: sentence vector describing a user's preference

Maximum marginal relevance: Example



For the first sentence,
MMR simply picks the sentence with **the highest similarity to the profile vector**.

Maximum marginal relevance: Example



For the next sentence(s),
the selected vector needs to be:

- **similar** to the profile
- **dissimilar** to the already-picked sentence(s)

This vector has the same distance to the centroid, but is closer to the already-picked vector!

Advantages and downsides of extractive summarization via MMR

- **Simple and fast** algorithm.
- Can be adjusted to **user preferences** via the profile vector.
- **Lack of semantics and cohesion** in selected sentences.
  - Sentences might not be linked correctly
  - "Dangling anaphora"; e.g. starting a summary with *He said yesterday that ...*
- Extracted sentences can be **longer than average**.
  - Depends entirely on sentence lengths in the original document

El-Kassas et al. (2020)

Important concepts

- extractive summarization
- sentence embeddings, sentence transformers
- maximum marginal relevance (MMR)
- profile vector

Large Language Models

3

Reminder: Language modelling

- What is the **probability** of a **sequence of words**?

$$p(\text{“I like books”}) > p(\text{“books I like”})$$
$$p(\text{“my comfort food is pizza”}) > p(\text{“my comfort food is chairs”})$$

- BERT models are trained on the **masked language modelling** (MLM) objective.

Kenya's athlete broke the world ***[MASK]*** *in long jump.*

Causal language modelling

1. *Berlin is the capital of* ***Germany*** — WORLD KNOWLEDGE
2. *Kenya's long-distance runner broke the world* ***record*** — LEXICAL KNOWLEDGE
3. *I almost fell asleep because this movie was so* ***boring*** — SENTIMENT
4. *If Alice is Bob's daughter, Bob is Alice's* ***father*** — SEMANTIC RELATION
5. *Yesterday we met the new* ****sees/*the/*because/...*** — SYNTACTICAL CONSTRAINTS

- In **causal language modelling**, we strictly predict from left to right.
  - ...but we still capture lots of different types of knowledge this way!

Masked language models are encoder models

- BERT is an example of an **encoder model**.
  - outputs are "encoded" vector representations
- Processes the **entire input sequence** before making predictions.
- Easily adaptable to sequence labelling or classification tasks.

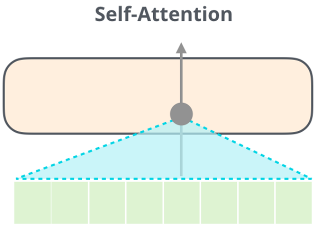


Illustration: Jay Alammar

Causal language models are (mostly) decoder models

- GPT-3 is an example of a **decoder model**.
  - "decodes" input into a *sequential output*
- Strictly **predicts the next word**.
- **Autoregressive**: predicted words are appended to the input.

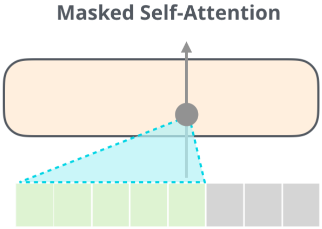
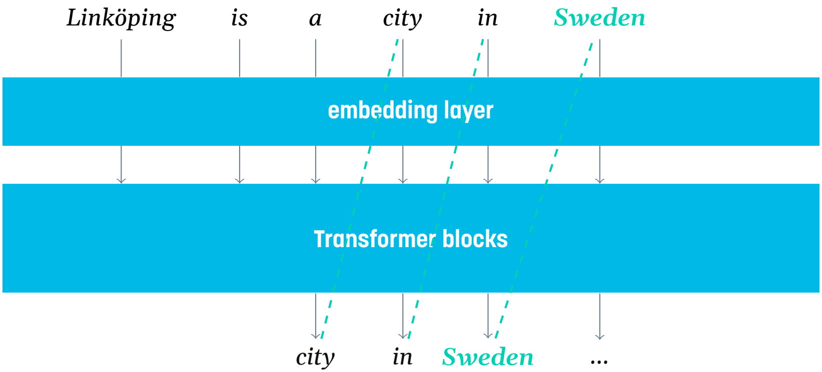


Illustration: Jay Alammar

Autoregressive decoder models



Prompting for text generation

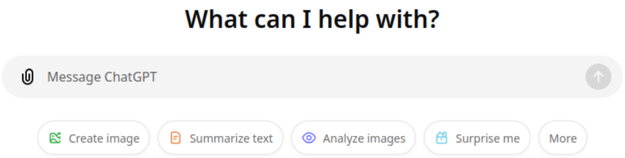
- Autoregressive decoder models are **text generation models**.
- With these models, we can **recursively sample** from their learned probability distribution to generate text:

> *The boiling point of water is* ***100 degrees Celsius or 212 degrees Fahrenheit at standard atmospheric pressure .***

From language models to assistant models

- Recent, state-of-the-art LLMs are almost always **decoder models**.
  - This means they are, at their core, **next word predictors**.

- However, products like ChatGPT function more like **AI assistants**.



Next word prediction is not sufficient

- How might a next word predictor continue the following prompt?

Should one discipline one's children by smacking them?
This question has long been debated without reaching agreement. We could settle it by a controlled manipulative experiment...

No incentive for the model to generate text that is helpful, ethical, truthful, etc.

- The language modelling objective trains the model to produce text that "looks similar" to its training data.

Source: The Guardian

Instruction fine-tuning

Idea

After training our language model on next word prediction, we **fine-tune it** on text that contains **instructions & the desired response**.

Description	*In this task, you're given an open-domain question. Your task is to provide an answer to the given question. There is only one unique answer for each question. Your answer should be short, and refer to an entity, date, number, etc. Do not put your answer in the context of a sentence.*
Input	*What does the DC in Washington DC stand for?*
Output	***District of Columbia***

Prompt templates

- Instruction fine-tuning often introduces **prompt templates** to the model.

```
<|im_start|>system
You are an AI assistant that provides an answer to the given question.
There is only one unique answer for each question. Your answer should
be short, and refer to an entity, date, number, etc. Do not put your
answer in the context of a sentence. <|im_end|>

<|im_start|>user
What does the DC in Washington DC stand for? <|im_end|>

<|im_start|>assistant
District of Columbia <|im_end|>
```

Data for IFT

Figure: Wang et al. (2022)
Data: Natural Instructions v2

Model alignment

- Instruction fine-tuning is way to **align** an LLM to **human preferences**.
  - Preferences can be helpfulness, truthfulness, etc.
  - Still uses next word prediction as its training objective!
- Models like ChatGPT add even more techniques to improve the alignment further.
  - *e.g.* reinforcement learning from human feedback (RLHF)
- This way, we go from a **language model** to an **assistant model**.

The size of modern LLMs

- The "size" of LLMs is typically expressed in **number of trainable parameters**.
- A direct consequence of that number is **how much (GPU) memory** is required to run the model.
  - Llama-2 7B: 10GB VRAM
  - Llama-2 13B: 24GB VRAM *(e.g. RTX 3090)*
  - Llama-2 70B: 80GB VRAM *(e.g. A100)*

model	year	params
BERT	2018	340 M
GPT-2	2019	1.5 B
GPT-3	2020	175 B
Gopher	2021	280 B
PaLM	2022	540 B
GLaM	2022	1,200 B
Llama-2	2023	70 B
Zephyr	2023	7 B

Hardware requirements for LLMs

- **1B–7B parameter models** can usually run on consumer-grade GPUs.
  - **Llama 3.2** models were released in 1B and 3B versions.
- **Tiny models** can even be run without a GPU, but may give worse results.
  - **SmolLM2** models come in 135M, 360M, and 1.7B versions.
- **Quantization** is currently a popular technique for reducing the size and memory requirements of any model.
  - Reduces the floating-point precision of the model parameters.
  - **5-bit Zephyr 7B Alpha** runs on CPU and ~8 GB RAM.

Tools for running LLMs locally

- **Huggingface Transformers** is probably the most versatile Python framework for working with LLMs locally.
  - Can also train and fine-tune models
  - Very limited support for quantized models
- **llama.cpp** is best for loading quantized models from Python.
- **Ollama** provides an API and command-line interface for LLMs.
- **LM Studio** provides a graphical interface for LLMs.

Important concepts

- causal language modelling
- encoder vs. decoder models
- autoregressive models
- instruction fine-tuning
- prompt templates

Abstractive Summarization

4

Abstractive summarization

- Produces summaries by **paraphrasing** text and **generating new sentences**.
  - It "abstracts away" from the original document.
  - Can result in better, more natural-sounding summaries.

Original text

The difficult romance gives the author a chance to discuss the theme of 'the light of reason vs. the darkness of passions.'

Abstractive summary

The review highlights the themes of reason vs. passion.

Jim Fonseca on Goodreads

Models for abstractive summarization

- Abstractive summarization is a **text generation** problem.
  - The output is now a sequence of tokens of arbitrary length!
- We can use **encoder–decoder models** for text generation.
  - Same architecture that is used for *e.g.* machine translation.
  - Examples include **BART** and **mT5**.
- Here, we focus on using **instruction fine-tuned LLMs**.

Abstractive summarization with LLMs

Idea

We can use the LLM's **prompt template** to **instruct it** to perform the task of abstractive summarization.

```
<|im_start|>user
Generate an abstractive summary of the text below. <|im_end|>

<|im_start|>input
Class divisions and changing values in Japan due to western influence
are major themes in the book Spring Snow by Yukio Mishima. The main
character is the son of a very wealthy family. … <|im_end|>

<|im_start|>assistant
This review of the book Spring Snow highlights the themes of …
```

Advantages and downsides of abstractive summarization with LLMs

- Typically **much better quality** of the resulting summaries.
- Easy to adjust to **user preferences** since instructions are given in natural language.
- **Less control** over the output.
  - Some prompt tweaking may be necessary to get the desired behaviour.
  - Model "hallucination," i.e. generation of wrong/incorrect output, is hard to detect.
- Much **higher compute requirements** than other techniques.
  - *But:* Can use LLMs with fewer parameters or quantizations

Evaluation of Text Summarization

5

Evaluating text summaries

- Evaluating the quality of generated **free-form text** is hard!

- There are many different aspects to a generated summary:

  1. *Is it **factually accurate**?*
  2. *Is it **informative**, i.e., does it include enough information?*
  3. *Is it **not redundant**, i.e., does it not repeat information?*
  4. *Is it **coherent**?*
  5. *Is it **fluent and well-written**?*

ROUGE: Recall-Oriented Understudy for Gisting Evaluation

- ROUGE is a set of metrics comparing a **system output** with one or more **references**.

Reference

Harry Potter star Daniel Radcliffe gets £20 M fortune as he turns 18 Monday. Young actor says he has no plans to fritter his cash away.

System output

Daniel Radcliffe, the Harry Potter star, gains access to a reported £20 million fortune as he turns 18, but insists he won't indulge in extravagant spending.

Example from CNN/DailyMail dataset and ChatGPT

ROUGE-N

- **ROUGE-*n*** counts the ***n*-gram overlap**.

Reference

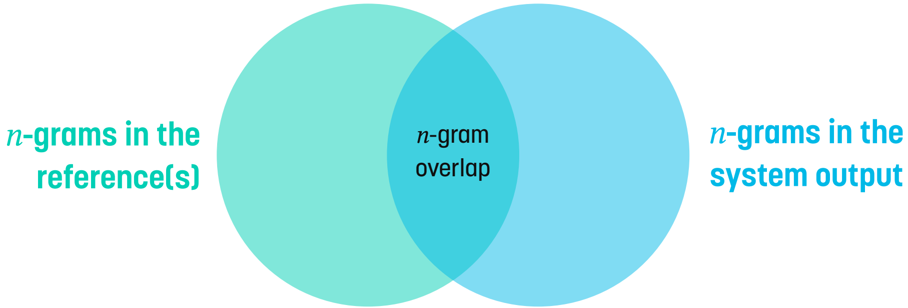
Harry Potter star Daniel Radcliffe *gets £20 M* ***fortune as he turns 18*** *Monday. Young actor says he has no plans to fritter his cash away.*

System output

Daniel Radcliffe*, the* ***Harry Potter star****, gains access to a reported £20 million* ***fortune as he turns 18****, but insists he won't indulge in extravagant spending.*

- Overlapping bigrams: *(Harry, Potter), (Potter, star), (Daniel, Radcliffe), (fortune, as), (as, he), (he, turns), (turns, 18)*

ROUGE-N: Precision and recall



$$P = \frac{|\text{references} \cap \text{system}|}{|\text{system}|} \qquad R = \frac{|\text{references} \cap \text{system}|}{|\text{references}|}$$

- ROUGE was originally defined as the **recall**, but in practice **F1-score** is often used.

ROUGE-L and ROUGE-S

- **ROUGE-L** considers the **longest common subsequences**.
- **ROUGE-S** considers **skip-bigrams**: pairs of words with "skipped" words in-between.

Reference

Harry Potter star Daniel Radcliffe gets ***£20*** *M* ***fortune*** *as he turns 18 Monday. Young actor says he has no plans to fritter his cash away.*

System output

Daniel Radcliffe, the Harry Potter star, gains access to a reported ***£20*** *million* ***fortune*** *as he turns 18, but insists he won't indulge in extravagant spending.*

Limitations of ROUGE

Reference

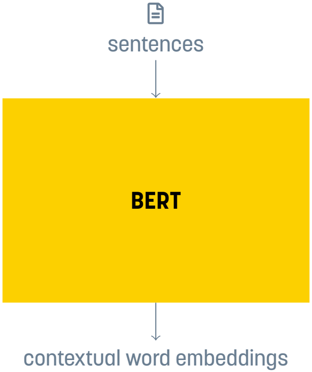
*Harry Potter star Daniel Radcliffe gets £20 M fortune as he turns 18 Monday. Young actor says **he has no plans to fritter his cash away**.*

System output

*Daniel Radcliffe, the Harry Potter star, gains access to a reported £20 million fortune as he turns 18, but insists **he won't indulge in extravagant spending**.*

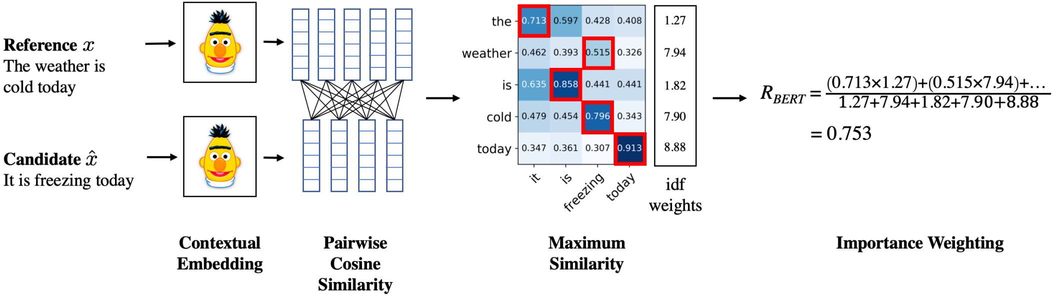
Word-level measures cannot account for **paraphrases or synonyms**.

BERTScore



- One popular alternative metric is BERTScore.
- **Compares the similarity of embeddings** between the reference and the system output.
  - Intuitively, while ROUGE performs simple *string matching*, this should compare the actual *meaning* of words.

Illustration of BERTScore



Zhang et al. (2020)

Which evaluation metric to use?

ROUGE

- 👍 easy and fast to compute
- 👎 only considers exact string matching

BERTScore

- 👍 considers *semantic* similarity
- 👎 scores have no clear interpretation

- Both of these metrics require a **reference** text to compare with.
  - Not always easy to obtain
  - *Reference-free* metrics have been proposed, e.g. GPTScore
- In many cases, **human evaluation** is still the most accurate solution!

Important concepts

- coherence, fluency, redundancy
- system output vs. references
- ROUGE metrics, ROUGE-n
- BERTScore

LINKÖPING UNIVERSITY