# Xử lý ngôn ngữ tự nhiên

Nhóm 6

# Nội dung

1. Đề tài
2. Phương pháp
3. Đánh giá
4. Hướng phát triển
5. Tài liệu tham khảo
6. Demo

# Đề tài

Tên đề tài: Tóm tắt văn bản tiếng Việt

Tự động tóm tắt sẽ là một trong những công nghệ quan trọng có thể giúp con người giảm thiểu thời gian đọc email và thông tin, kiến thức mới để dành thời gian cho các công việc khác, mà vẫn có thể nắm bắt được gãy gọn những nội dung của nó.

# Đề tài

Đầu vào: Một văn bản tiếng Việt (đơn văn bản) với nhiều đoạn văn

Đầu ra: Một đoạn văn bản tóm tắt với số lượng câu chọn trước (ít hơn số lượng câu của văn bản ban đầu)

# Đề tài

Thách thức

- Lượng dữ liệu lớn
- Ngữ nghĩa phức tạp
- Giữ được thông tin quan trọng
- Mạch lạc nội dung
- Trùng lặp nội dung
- Sắp xếp thông tin

# Phương pháp

Tiền xử lý → Word2Vec / BERT → Phân cụm

Mô hình 1: Sử dụng Word2Vec + KMeans

Mô hình 2: Sử dụng BERT + KMeans

# Phương pháp

Mô hình 1: Sử dụng Word2Vec [1]

Tiền xử lý

→ Chuyển dạng viết thường và loại bỏ khoảng trống thừa

→ Tách câu

[1] https://github.com/Kyubyong/wordvectors

# Phương pháp

Mô hình 1: Sử dụng Word2Vec

Word2Vec → chuyển các câu sang dạng vector

Kích thước vector là 100 chiều, corpus là 74MB, vocabulary là 10087 từ

Tách từ (pyvi) → vector hóa các từ → cộng tổng → vector cả câu → danh sách vector

# Phương pháp

Mô hình 1: Sử dụng Word2Vec

KMeans [2] → phân cụm các vector

$$n_{cluster} = \begin{cases} 1 \text{ nếu } ratio * len(sentences) < 1 \\ ratio * len(sentences) \text{ trong trường hợp còn lại} \end{cases}$$

Xây dựng đoạn tóm tắt

→ với mỗi cụm, chọn ra một câu duy nhất đại diện cho cụm (là câu có khoảng cách gần với centroid của cụm nhất)

→ sắp xếp thứ tự cụm

[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

# Phương pháp

Mô hình 2: Sử dụng BERT [3]

Tiền xử lý:

→ Thay thế ký tự xuống dòng thành ký tự khoảng trống

→ Loại bỏ khoảng trống thừa

[3] https://arxiv.org/abs/1906.04165

# Phương pháp

Mô hình 2: Sử dụng BERT

BERT → chuyển câu sang vector

→ sử dụng neuralcoref → xây dựng data pineline → giải quyết vấn đề coreference → danh sách các câu được tách

→ sử dụng mô hình BERT base với sub-layer là 2, kích thước embedding vector là 768, số head trong lớp self-attention là 12

→ giữa các lớp ẩn, thực hiện pooling với average pooling

→ danh sách các vector

# Phương pháp

Mô hình 2: Sử dụng BERT

KMeans → phân cụm các vector

Xây dựng đoạn tóm tắt

→ với mỗi cụm, chọn ra một câu duy nhất đại diện cho cụm (là câu có khoảng cách gần với centroid của cụm nhất)

→ sắp xếp thứ tự, giữ nguyên thứ tự của câu trong văn bản đầu vào

# Đánh giá

Dữ liệu đánh giá

→ gồm các văn bản thuộc các chủ đề khác nhau:

- Khoa học công nghệ (25)
- Chính trị (31)
- Khoa học giáo dục (22)
- Kinh tế (53)
- Văn hóa (34)
- Xã hội (35)

# Đánh giá

Độ đo đánh giá

Rouge

BERT_score

→ Precision, Recall

→ F1 Score

# Đánh giá

So sánh

| | Rouge | BERT_score |
|---|---|---|
| Mô hình 1 | | |
| Mô hình 2 | | |

# Demo

Công nghệ sử dụng

- HTML, JavaScript, Bootstrap
- Flask

Tính năng

- Tóm tắt nhanh văn bản
- So sánh mô hình

# Hướng phát triển

Tiền xử lý được các trường hợp phức tạp hơn của văn bản (dấu câu trong tên nước ngoài, dấu câu trong số như 3.000, …)

Tối ưu các tham số của mô hình

# Tài liệu tham khảo

[1] https://github.com/Kyubyong/wordvectors
[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[3] https://arxiv.org/abs/1906.04165

# Thank you !