MỘT PHƯƠNG PHÁP DỰA TRÊN LUẬT ĐỂ CHUYỂN ĐỔI VĂN BẢN TIẾNG VIỆT VỀ DRS (DISCOURSE REPRESENTATION STRUCTURE)

Trần Trung[1], *Nguyễn Tuấn Đăng*[1]

Tóm tắt

Biểu diễn ngữ nghĩa văn bản là bước quan trọng đầu tiên của một hệ thống tóm lược văn bản (K. S. Jones [25, 26]) theo hướng tiếp cận tóm lược trừu tượng. Quy trình tóm lược văn bản theo cách tiếp cận này phải thực hiện các bước xử lý chính: (i) Chuyển đổi văn bản đầu vào thành một dạng biểu diễn ngữ nghĩa; (ii) Chuyển đổi dạng biểu diễn ngữ nghĩa này sang một dạng biểu diễn đầu ra; (iii) Tạo sinh tóm lược từ biểu diễn đầu ra. Trong bài báo này, chúng tôi trình bày một phương pháp mới để thực hiện bước xử lý thứ nhất trong quy trình trên. Phương pháp này được chúng tôi áp dụng vào mô hình hệ thống tóm lược văn bản theo hướng tiếp cận trừu tượng mà chúng tôi đã đề xuất trong những nghiên cứu trước. Phương pháp này được nghiên cứu để áp dụng cho những đoạn văn bản ngắn bao gồm từ 2 đến 5 câu tiếng Việt có sự liên kết về đại từ hồi chỉ giữa một số câu. Đầu ra của phương pháp là cấu trúc Discourse Representation Structure (DRS) để biểu diễn ngữ nghĩa của đoạn văn bản đầu vào. Phương pháp chuyển đổi một đoạn văn bản thành DRS gồm có ba giai đoạn: (i) Phân tích cấu trúc nông của các câu nguyên bản với bộ công cụ OpenNLP (Tokenizer, POS Tagger và Chunker); (ii) Ánh xạ từng cấu trúc nông về dạng cấu trúc bao gồm những ngữ đoạn chứa nội dung chính, đồng thời tái tạo lại câu dựa trên những ngữ đoạn này; (iii) Tạo lập cấu trúc DRS của đoạn văn bản kết quả ở giai đoạn thứ hai.

Text meaning representation is the first important step in a text summarization system (K. S. Jones [25, 26]) following the abstractive direction. The summarization process following this direction has three main performing steps: (i) Transform the input text into a semantic representation form; (ii) Transfom this semantic form into an output form; (iii) Generate the summary from the output form. In this article, we present a new method for performing the first step in the above process. This method is applied in the abstractive summarization model which was proposed in previous researches. We apply this method for short paragraphs composing from 2 to 5 Vietnamese sentences which have the anaphoric pronoun relationships between some sentences. The output of the method is Discourse Representation Structure (DRS) presenting the semantic of the input paragraph. The transforming method includes three stages: (i) Parse the shallow structures of the original sentences by OpenNLP tools (Tokenizer, POS Tagger and Chunker); (ii) Map the shallow structure to reduced structure composing main phrases, and re-create the sentence from these phrases. (iii) Build the DRS of the result paragraph from stage two.

Từ khóa

Discourse Representation Structure, Đại từ hồi chỉ, Gán nhãn từ loại, Gán nhãn ngữ đoạn, Phân tích cú pháp.

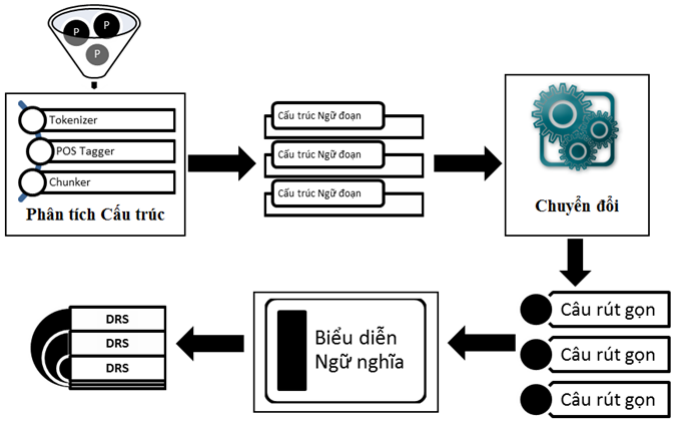
[1] Khoa Khoa học Máy tính, Trường Đại học Công nghệ thông tin – ĐHQG Tp. Hồ Chí Minh.

1. Giới thiệu

Nhìn chung, trong lĩnh vực Tóm lược văn bản, một hệ thống tóm lược thông thường theo K. S. Jones [25, 26] cần có ba thành phần cơ bản để xử lý ba tác vụ quan trọng: (i) Biểu diễn được ngữ nghĩa của văn bản ban đầu; (ii) Tạo dựng được cấu trúc biểu diễn mới phù hợp từ biểu diễn thứ nhất; (iii) Tạo sinh được văn bản tóm lược từ cấu trúc biểu diễn mới. Ba tác vụ này càng trở nên quan trọng đối với những nghiên cứu theo hướng tiếp cận tóm lược trừu tượng ([2, 7, 11, 13, 19, 24, 25, 26, 33]).

Trong giải pháp chung về tóm lược – tạo sinh câu được trình bày trong những nghiên cứu trước ([44, 45, 46]), với nền tảng ý tưởng của hướng tiếp cận tóm lược trừu tượng "abstraction" ([11]), chúng tôi trình bày những phương pháp và kỹ thuật khác nhau để tóm lược các nhóm đoạn văn bản tiếng Việt gồm các câu đơn giản có những đặc điểm khác nhau bằng cách tạo sinh những câu tiếng Việt mới thỏa mãn tính tự nhiên trong giao tiếp tiếng Việt thông thường. Ý tưởng chính xuyên suốt trong các nghiên cứu này ([44, 45, 46]) bao gồm các tiến trình xử lý chính: (i) thay thế những đại từ hồi chỉ bằng đối tượng được liên hệ tương ứng và biểu diễn ý nghĩa của đoạn văn bản tiếng Việt ban đầu bởi một cấu trúc biểu diễn ngữ nghĩa; (ii) dựa trên những dạng quan hệ được định nghĩa trước để xác định những quan hệ nội tại và liên câu trong quá trình phân tích cấu trúc biểu diễn ngữ nghĩa; (iii) tạo sinh cấu trúc cú pháp của các câu tiếng Việt mới với giải thuật phù hợp theo các quan hệ tìm được và ngữ cảnh thực tế ban đầu; (iv) hoàn chỉnh đoạn văn bản tiếng Việt mới.

Nội dung chính của nghiên cứu này là tập trung vào vấn đề tiền xử lý cho giải pháp chung bên trên của chúng tôi ([44, 45, 46]): đề xuất một phương pháp chuyển đổi những đoạn văn bản tiếng Việt có cấu trúc phức tạp về dạng rút gọn chỉ bao gồm những câu đơn giản chứa nội dung chính và có thể biểu diễn ngữ nghĩa theo mô hình DRS (Discourse Representation Structure). Chúng tôi dựa theo lý thuyết DRT (Discourse Representation Theory [15, 29, 35]) trong những nghiên cứu trước ([44, 45, 46]) để sử dụng DRS làm cấu trúc biểu diễn ngữ nghĩa cho những đoạn văn bản gồm những câu đơn giản. Mô hình tổng quan giải pháp chuyển đổi được trình bày trong hình 1.



Hình 1. *Mô hình tổng quan giải pháp chuyển đổi đoạn văn bản tiếng Việt về dạng biểu diễn ngữ nghĩa.*

Trong mô hình chuyển đổi được thể hiện trong hình 1, có ba giai đoạn thực hiện chính:

1) *Giai đoạn thứ nhất được thực hiện trong bộ Phân tích cấu trúc.* Đầu vào của bộ Phân tích cấu trúc là một đoạn văn bản tiếng Việt được phân tách thành từng câu riêng biệt. Đầu ra của bộ Phân tích cấu trúc là cấu trúc nông của các câu này (theo quan điểm của nghiên cứu này, là những câu trong đó các từ vựng và ngữ đoạn được gán nhãn phù hợp). Chúng tôi sử dụng các bộ công cụ của Apache OpenNLP (Tokenizer, POS Tagger, Chunker)[1,2] để tiến hành việc huấn luyện ngữ liệu và phân tích các câu đầu vào. Để phù hợp với giải pháp chung về tóm lược ([44, 45, 46]), chúng tôi định nghĩa những nhãn từ vựng mới dựa trên tư tưởng phân loại câu theo nghĩa biểu hiện trong lý thuyết Ngữ pháp Chức năng ([17, 30]).
2) *Giai đoạn thứ hai được thực hiện trong bộ Chuyển đổi.* Đầu vào của bộ Chuyển đổi là những cấu trúc nông của các câu được chuyển đến từ bộ Phân tích cấu trúc. Đầu ra của bộ Chuyển đổi là những câu tiếng Việt ở dạng rút gọn có cấu trúc đơn giản. Ở bước thứ nhất trong bộ Chuyển đổi, chúng tôi đề xuất một giải thuật để ánh xạ cấu trúc phân tích ngữ đoạn của từng câu ở pha thứ nhất về dạng cấu trúc bao gồm những ngữ đoạn chứa nội dung chính. Ở bước tiếp theo, chúng tôi tái tạo những câu tiếng Việt dựa trên dạng cấu trúc mới này.
3) *Những câu tiếng Việt rút gọn được tái tạo từ bộ Chuyển đổi sẽ trở thành đầu vào của bộ Biểu diễn ngữ nghĩa trong giai đoạn thứ ba.* Chúng tôi áp dụng những phương pháp và kỹ thuật trong những nghiên cứu trước ([44, 45, 46]) để tạo lập cấu trúc DRS biểu diễn ngữ nghĩa đoạn văn bản.

Nội dung chính của bài báo được phân chia thành các phần như sau. Trong Phần 2, chúng tôi trình bày việc sử dụng những mô hình biểu diễn ngữ nghĩa khác nhau trong những nghiên cứu có liên quan về tóm lược theo hướng tiếp cận trừu tượng. Phần 3 trình bày tổng quan về biểu diễn ngữ nghĩa bằng cấu trúc DRS. Trong Phần 4, chúng tôi trình bày bộ nhãn được xây dựng phù hợp mục tiêu nghiên cứu và việc phân tích cấu trúc nông trong pha thứ nhất, giải thuật để chuyển đổi từ cấu trúc nông về dạng rút gọn. Thử nghiệm và so sánh với một số phương pháp tóm lược được trình bày trong Phần 5. Cuối cùng, trong Phần 6, chúng tôi trình bày kết luận và đề xuất một số điểm cải tiến trong tương lai.

2. Những nghiên cứu liên quan

Tác vụ biểu diễn ngữ nghĩa văn bản ban đầu được xem là rất quan trọng trong các nghiên cứu tóm lược theo hướng tiếp cận tóm lược trừu tượng. Nhiều tác giả đề xuất việc áp dụng những mô hình biểu diễn ngữ nghĩa khác nhau trong hệ thống nhằm tạo sinh văn bản đầu ra rút gọn, mang đầy đủ thông tin và mang tính tự nhiên nhất đối với sự tri nhận của con người.

[1]https://opennlp.apache.org/

[2]https://opennlp.apache.org/documentation/1.6.0/manual/opennlp.html

S. M. Harabagiu và F. Lacatusu [43] đã xây dựng một cấu trúc mẫu là những mẩu văn bản nhỏ để biểu diễn những văn bản cho trước và tạo sinh văn bản tóm lược. Các tác giả áp dụng những quy tắc trong một hệ thống trích xuất thông tin để trích xuất những thông tin từ đa văn bản. Những thông tin này được sử dụng để lấp đầy những mẫu và tạo sinh những tóm lược đa văn bản có tính mạch lạc và cung cấp nhiều tin tức.

Một hướng tiếp cận biểu diễn ngữ nghĩa bằng ontology mờ được C. S. Lee và cộng sự [8] áp dụng, giúp cho việc xử lý những dữ liệu hay thay đổi và có thể tóm lược tốt những văn bản trên những trang điện tử có cấu trúc dữ liệu riêng. Cùng ý tưởng áp dụng mô hình ontology, C. F. Greenbacker [6] đề xuất một nền tảng để tạo sinh những tóm lược theo hướng tiếp cận trừu tượng với ba bước chính: (i) Xây dựng một biểu diễn mô hình ngữ nghĩa bằng ontology cho những nội dung của những văn bản đa thể hiện; (ii) Một metric đánh giá những khái niệm trong ontology với nhiều yếu tố như tính đầy đủ của những thuộc tính, số lượng mối quan hệ với những khái niệm khác,...; (iii) Tạo tóm lược với những khái niệm quan trọng nhất.

P. E. Genest và G. Lapalme [36] tạo sinh tóm lược từ những mục tin tức (information item) biểu diễn trừu tượng của những văn bản nguồn. Những mục tin này được xây dựng trong thành phần đầu tiên trong hệ thống tóm lược của các tác giả, phân tích cú pháp của văn bản nguồn và trích xuất những chủ đề của động từ và đối tượng. Tiếp đó, họ tạo sinh câu mới và lựa chọn câu bằng cách đánh giá những câu được tạo sinh với những điểm số phù hợp. Cuối cùng, họ tạo sinh tóm lược bằng cách kết hợp những câu được tạo sinh có điểm cao nhất với những thông tin về ngày tháng và địa điểm để tạo thành tóm lược hoàn chỉnh. Mặc dù tóm lược ngắn ngọn, có tính mạch lạc, giàu thông tin và ít rườm rà, vẫn còn đó những giới hạn của phương pháp tiếp cận này: (a) Có thể bỏ qua một số những mục tin tức mà có thể khó khăn trong việc tạo thành những câu có ý nghĩa và đúng ngữ pháp; (b) Trong thành phần lấy mục tin, nếu việc phân tích thành cây cú pháp không chính xác, như vậy chất lượng ngôn ngữ học của tóm lược là không cao.

Một nghiên cứu sử dụng đồ thị ngữ nghĩa được I. F. Moawad và M. Aref [18] áp dụng để biểu diễn ngữ nghĩa của văn bản nguồn. Các tác giả sẽ lọc bớt thông tin trên đồ thị bằng việc sử dụng những quy tắc heuristic. Từ đó, tóm lược tóm tắt sẽ được tạo sinh với kết quả khá súc tích, mạch lạc và ít rườm rà. Tuy nhiên, phương pháp này thiếu những kiến thức trong những lý thuyết ngôn ngữ học. Do vậy, tóm lược có thể không hoàn toàn đúng ngữ pháp và không mang tính tự nhiên trong ngôn ngữ được áp dụng.

R. Barzilay và cộng sự [40] phát triển một phương pháp trộn câu trong ngữ cảnh của tóm lược đa văn bản với những bước xử lý chính:

- Nhận biết những thông tin chung. Trong bước này, những thông tin được chia sẻ giữa hai câu nguồn sẽ được nhận biết. Trước tiên, từng câu sẽ được biểu diễn cấu trúc cú pháp thành cây phụ thuộc. Sau đó, những cây này sẽ được so khớp. Những cây phụ thuộc được thiết kế tốt với những thông tin của từng từ vựng trong câu được lưu trữ tại mỗi nút.

- So khớp những cây phụ thuộc. Trong bước này những cây phụ thuộc sẽ được kết hợp để tạo thành một cây duy nhất. Cây được tạo thành sẽ được tiếp tục điều chỉnh thông qua hai bước: (i) Làm giàu cây bằng cách thêm vào những đường đi và cây con với điều kiện chúng xuất hiện đủ nhiều trong những câu ban đầu; (ii) Tỉa cây bằng cách loại bớt những cây con không xuất hiện nhiều trong các cây đầu vào và không có tầm quan trọng về ngữ pháp được quy định trước.
- Tuyến tính hóa cây là tác vụ cuối cùng trong hệ thống trộn câu theo phương pháp này. Cây phụ thuộc đầu ra sẽ được sử dụng làm nền tảng tạo sinh câu tóm lược. Những chuỗi câu ứng viên khác nhau sẽ được tạo sinh và được xếp hạng dựa theo một mô hình ngôn ngữ tri-gram.

E. Krahmer và cộng sự [12] mở rộng tiếp cận của Barzilay và McKeown [40] để thực hiện trộn câu trong ngữ cảnh của các hệ thống hỏi đáp. Họ không chỉ thực hiện trộn giao nhau mà còn thực hiện trộn hỗn hợp. Thực hiện tương tự Barzilay và McKeown [40] cho ba thành phần chính của một hệ thống trộn câu: (i) Ở thành phần sắp hàng, đề xuất chiến lược để hiểu thêm làm thế nào những từ vựng và ngữ đoạn trong các câu đầu vào liên hệ với nhau, trong đó có sự chồng chéo thông tin, diễn giải thông tin theo các cách khác nhau, sự kế thừa thông tin,...; (ii) Ở thành phần trộn, xác định trước dạng trộn là hỗn hợp hay giao nhau để quyết định trộn cây phụ thuộc; (iii) Cuối cùng, xác định sự tuyến tính hóa tốt nhất với một mô hình ngôn ngữ.

K. Filippova và M. Strube [22, 23] đề xuất phương pháp trộn câu không giám sát, bắt đầu với việc sắp hàng những cây phụ thuộc của các câu có quan hệ với nhau để tạo thành một đồ thị phụ thuộc. Tiếp đến, họ sử dụng lập trình tuyến tính số nguyên (integer linear programming) để nén đồ thị này và tạo thành cây mới. Các đồng đối số trong câu kết quả được kiểm tra tính tương thích về cú pháp và ngữ nghĩa.

Phương pháp sử dụng đồ thị từ vựng được K. Fillipova [21] đề xuất với ưu điểm không sử dụng tài nguyên bên ngoài. Phương pháp này bao gồm việc sử dụng một đồ thị từ vựng của các câu cần được trộn và nén, từ đó xác định đường dẫn ngắn nhất chứa các thông tin chung để tạo thành tóm lược. Những kỹ thuật chính được sử dụng trong phương pháp này là tách từ và gán nhãn từ vựng.

Boudin và Morin [14] cải tiến phương pháp của K. Fillipova [21] bằng cách đề xuất phương pháp đánh giá lại dựa theo việc trích xuất những cụm từ khóa và tạo ra những câu nén có nhiều thông tin và được nhấn mạnh hơn.

3. Biễn diễn ngữ nghĩa văn bản theo cấu trúc Discourse Representation Structure

Lý thuyết Discourse Representation Theory (DRT) được giới thiệu trong [15, 29, 35] với ý tưởng cơ bản: Một đoạn văn bản ngôn ngữ tự nhiên sẽ được biểu diễn trong một ngữ cảnh của một cấu trúc biểu diễn, cấu trúc này được gọi là Discourse Representation Structures (DRS). Kết quả của tiến trình xử lý một phần đoạn văn bản trong ngữ cảnh của cấu trúc biểu diễn A sẽ là một cấu trúc biểu diễn mới A'.

Theo [15, 29, 35], một cấu trúc DRS sẽ bao gồm một cặp danh sách có thứ tự <U, Con>, trong đó: (i) U là một danh sách những đánh dấu văn bản, hay còn có thể hiểu là những đối tượng của văn bản; (ii) Con là danh sách những điều kiện, hay có thể hiểu là những vị từ hay công thức mà những đối tượng trong danh sách U phải thỏa. Như một ví dụ, xét đoạn văn bản gồm 3 câu đơn giản trong Ví dụ 1. Đoạn văn bản này có cấu trúc DRS được trình bày với hai danh sách U và Con như trong Hình 2.

Ví dụ 1. "*Nhân học môn vẽ. Anh dùng bút chì. Nghĩa hỏi anh.*"

$\Longrightarrow$ Cấu trúc DRS với hai danh sách U và Con: (i) Danh sách U bao gồm những đối tượng: `1 - Nhân, 2 - môn vẽ, 3 - bút chì, 4 - Nghĩa`; (ii) Danh sách Con bao gồm những điều kiện: `tên(1,[nhân]), môn_vẽ(2), học(1,2), bút_chì(3), dùng(1,3), tên(4,[nghĩa]), hỏi(4,1).`

$\Longrightarrow$ Cấu trúc DRS được biểu diễn dưới dạng bảng:

[1, 2, 3, 4]
tên(1,[nhân])
môn_vẽ(2)
học(1,2)
bút_chì(3)
dùng(1,3)
tên(4,[nghĩa])
hỏi(4,1)

Hình 2. Cấu trúc DRS của đoạn văn bản "Nhân học môn vẽ. Anh dùng bút chì. Nghĩa hỏi anh."

Quá trình tạo dựng cấu trúc DRS được thực hiện tuần tự từng câu trong đoạn văn bản, cập nhật thông tin câu mới vào ngữ cảnh của những câu trước đó. Quá trình này đồng thời cũng giúp xác định đối tượng tiền ngữ phù hợp cho từng đại từ hồi chỉ xuất hiện ở các câu tiếp theo và cập nhật vào cấu trúc.

4. Phân tích cấu trúc nông văn bản Tiếng Việt ban đầu và chuyển đổi về dạng cấu trúc rút gọn

Trong phần này, chúng tôi trình bày tập nhãn Fn Tagset được xây dựng phù hợp mục tiêu nghiên cứu và phương pháp phân tích cấu trúc nông cũng như chuyển đổi về dạng cấu trúc rút gọn.

4.1. Tập nhãn Fn Tagset và quá trình phân tích cấu trúc nông

Nền tảng của việc xây dựng tập nhãn Fn Tagset là sự phân loại câu tiếng Việt dựa trên nghĩa biểu hiện của lý thuyết ngôn ngữ học Ngữ pháp Chức năng ([17, 30]). Trong những loại câu này, nghiên cứu tập trung xem xét những dạng câu sau: (i) Câu chỉ hành

động, gồm hai loại con "chuyển tác (tác động – tạo diệt)" và "phi chuyển tác (di chuyển – không di chuyển)"; (ii) Câu chỉ quá trình, gồm hai loại con "chuyển tác (tác động – tạo diệt)" và "phi chuyển tác (chuyển biến – sinh diệt)"; (iii) Câu chỉ trạng thái, gồm hai loại con "tính chất (phẩm chất – tính khí)" và "tình trạng (thể chất – tâm trạng)".

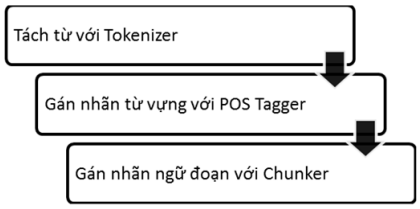
Dựa trên những dạng câu trên, chúng tôi xây dựng tập nhãn Fn Tagset bao gồm những loại nhãn chính: (i) Nhãn câu, mệnh đề, ngữ đoạn - Chúng tôi áp dụng những nhãn câu và mệnh đề trong bộ nhãn của dự án Penn-Treebank [5]; (ii) Nhãn từ vựng - Chúng tôi định nghĩa nhãn từ vựng cho những từ loại chính bao gồm {tất cả các loại đối tượng được biểu diễn bởi danh từ riêng hay danh từ chung trong câu; tất cả các loại hành động, quá trình hay trạng thái được biểu diễn bởi động từ hay tính từ trong câu; tất cả các loại đại từ hồi chỉ} và áp dụng những nhãn từ vựng trong bộ nhãn của dự án Penn-Treebank [5] để gán nhãn cho những từ loại còn lại.

Toàn bộ nhãn từ vựng trong bộ nhãn Fn Tagset được phân loại cụ thể như sau:

- "Đối tượng": {"Người": ["Chung – General" → HUMG]; ["Cụ thể – Concrete" → HUMC]; ["Không thực – Non-Real" → HUMN]}; {"Động vật": ["Chung – General" → ANIG]; ["Cụ thể – Concrete" → ANIC]; ["Không thực – Non-Real" → ANIN]}; {"Tĩnh vật": ["Chung – General" → NANIG]; ["Cụ thể – Concrete" → NANIC]; ["Không thực – Non-Real" → NANIN]}; {"Phi vật thể": ["Chung – General" → NENTG]; ["Cụ thể – Concrete" → NENTC]}; {"Phi vật chất": ["Chung – General" → IMATG]; ["Cụ thể – Concrete" → IMATC]}; {"Hiện tượng": ["Chung – General" → PHEG]; ["Cụ thể – Concrete" → PHEC]}; {"Không gian": ["Chung – General" → SPAG]; ["Cụ thể – Concrete" → SPAC]; ["Không thực – Non-Real" → SPAN]}; {"Thời gian": ["Chung – General" → TIMG]; ["Cụ thể – Concrete" → TIMC]}
- "Hành động":
  - "Vô tác": ["Di chuyển – Moving" → ITMO]; ["Thay đổi tư thế cơ thể – Posture Changed" → ITPCH]; ["Tinh thần – Spiritual" → ITSP]; ["Ứng xử – Behaviour" → ITBE]; ["Tạo lập – Creation" → ITCR]; {"Biến đổi": ["Tạo lập – Make up" → ITTMA]; ["Màu sắc – Colour" → ITTCO]; ["Ánh sáng – Light" → ITTLI]; ["Âm thanh – Sound" → ITTSO]; ["Chuyển động – Motion" → ITTMO]}
  - "Chuyển tác": ["Tạo lập – Creation" → TCRE]; {"Biến đổi": ["Tạo lập – Make up" → TTMA]; ["Bề mặt – Surface" → TTSU]; ["Ánh sáng – Light" → TTLI]; ["Bên ngoài – Exterior" → TTEX]; ["Bên trong – Interior" → TTIN]; ["Giao tiếp – Contact" → TTCON]; ["Sở hữu – Possession" → TTPO]; ["Hoạt động – Operation" → TTOPE]; ["Trạng thái – State" → TTST]; ["Kích thước – Size" → TTSI]; ["Hình dáng – Shape" → TTSH]; ["Thời kỳ – Age" → TTAG]; ["Số lượng – Amount" → TTAM]; ["Màu sắc – Colour" → TTCOL]; ["Bổ sung – Accompaniment" → TTAC]; ["Chuyển động – Motion" → TTMO]}
- "Quá trình": ["Chuyển thái – State Changed" → PCSCH]; ["Chuyển vị – Position Changed" → PCPCH]; ["Nảy sinh – Emerge" → PCEME]; ["Hủy diệt – Destroy" → PCDES]
- "Trạng thái"

  - "Tính chất": {"Thể chất": ["Vật vô sinh – Non-Biotic Material" → PPBNB]; ["Vật hữu sinh – Biotic Material" → PPBBM]}; {"Tinh thần": ["Chung – General" → PPSGE]; ["Trí tuệ (Người) – Human Intelligence" → PPSHI]; ["Đạo đức (Người) – Human Ethical" → PPSHE]; ["Ứng xử (Người) – Human Behaviour" → PPHSB]; ["Cảm tính (Người) – Human Sensitivity" → PPSHS]}
  - "Tình trạng": {"Vật chất": ["Vật vô sinh – Non-Biotic Material" → STMNB]; ["Vật hữu sinh – Biotic Material" → STMBM]}; {"Tâm lý": ["Cảm giác – Sensation" → STPSE]; ["Tâm trạng – State Of Mind" → STPSO]}
  - "Cảm giác": ["Cảm thụ – Perceptive" → SEPER]; ["Tri nhận – Cognitive" → SECOG]; ["Khao khát – Desiderative" → SEDES]; ["Xúc cảm – Emotive" → SEEMO]
- "Đại từ hồi chỉ": {"Chính danh": ["Chỉ sự vật" → PRCNT]; ["Chỉ người" → PRCNH]}; {"Gốc danh từ + tính từ chỉ thị": ["Chỉ sự vật" → PRNDT]; ["Chỉ người" → PRNDH]}

Quá trình phân tích cấu trúc nông được thực hiện với các tiến trình thực thi sau:



Hình 3. *Ba tiến trình thực thi bộ Phân tích cấu trúc với các công cụ của Apache OpenNLP*[1,2]

Chúng tôi thực thi các tiến trình trong Hình 3 theo hai vòng xử lý chính sau:

1) *Vòng thứ nhất.* Huấn luyện ngữ liệu. Trong vòng này, dựa trên mô tả trong tài liệu của OpenNLP[1,2], chúng tôi thực hiện theo ba bước đối với từng tiến trình: (i) **Bước 1.** Tổ chức tập tin chứa dữ liệu huấn luyện; (ii) **Bước 2.** Đọc tập tin chứa dữ liệu huấn luyện; (ii) **Bước 3.** Thực hiện huấn luyện và tạo sinh model.
2) *Vòng thứ hai.* Phân tích cấu trúc ngữ đoạn câu tiếng Việt đầu vào. Trong vòng này, dựa trên mô tả trong tài liệu của OpenNLP[1,2], chúng tôi thực hiện theo ba bước để gán nhãn ngữ đoạn: (i) **Bước 1.** Tách từ trong câu sử dụng Tokenizer; (ii) **Bước 2.** Gán nhãn từ vựng sử dụng POS Tagger; (iii) **Bước 3.** Gán nhãn ngữ đoạn sử dụng Chunker.

Việc huấn luyện dữ liệu được thực hiện ở vòng thứ nhất như sau:

4.1.1. Huấn luyện ngữ liệu với Apache OpenNLP Tokenizer: Dựa trên mô tả trong tài liệu của OpenNLP[1,2], chúng tôi thiết lập một tập tin chứa đựng những câu tiếng Việt theo cấu trúc sau: (i) Mỗi dòng là một câu tiếng Việt; (ii) Các từ đơn hoặc các âm tiết của một từ ghép được cách nhau bởi một khoảng trắng; (iii) Các dấu phân tách như "," hay "." được phân cách với từ gần nhất bên trái bởi nhãn đặc trưng "<SPLIT>". Trong nghiên cứu này, chúng tôi xem các âm tiết của một từ ghép cũng là những từ đơn, và như vậy cũng được viết tách ra để huấn luyện. Như một ví dụ, xem xét đoạn văn bản

gồm hai câu tiếng Việt trong Ví dụ 2. Cặp câu này được định dạng trong tập tin huấn luyện như Hình 4:

Ví dụ 2. "*Đổng Ông hồi hộp mở tủ. Ông tái mặt đứng vịn thành tủ một lát.*"

```
đổng ông hồi hộp mở tủ<SPLIT>.
ông tái mặt đứng vịn thành tủ một lát<SPLIT>.
```

Hình 4. *Định dạng huấn luyện tách từ cho cặp câu trong Ví dụ 2*

4.1.2. Huấn luyện ngữ liệu với Apache OpenNLP POS Tagger: Dựa trên mô tả trong tài liệu của OpenNLP[1,2], chúng tôi thiết lập một tập tin chứa đựng những câu tiếng Việt theo cấu trúc sau: (i) Mỗi dòng là một câu tiếng Việt; (ii) Mỗi cặp âm tiết và nhãn được nối bởi dấu "_"; (ii) Hai cặp "âm tiết_nhãn" liên tiếp được cách nhau bởi khoảng trắng. Quy ước gán nhãn từ vựng được thực hiện như sau: (a) Đối với từ đơn - gán nhãn theo quy ước của OpenNLP[1,2] "từ vựng_nhãn"; (b) Đối với từ ghép (theo nghĩa rằng từ vựng có từ hai âm tiết trở lên) - bước (b1) định nghĩa một nhãn con của nhãn từ vựng tương ứng bằng cách thêm dấu "_" vào trước nhãn nguyên bản và bước (b2) gán nhãn từng âm tiết với nhãn con này tương tự gán nhãn từ đơn. Như một ví dụ, các từ vựng trong cặp câu trong Ví dụ 2 được gán nhãn trong tập tin huấn luyện như Hình 5

```
đổng__HUMC ông__HUMC hồi__STPSE hộp__STPSE mở_TTOPE tủ_NANIC ._.
ông_PRCNH tái_STMBM mặt_NANIC đứng_ITPCH vịn_TTOPE thành__NANIC
tủ__NANIC một_CD lát_TIMC ._.
```

Hình 5. *Định dạng huấn luyện gán nhãn từ cho cặp câu trong Ví dụ 2.*

4.1.3. Huấn luyện ngữ liệu với Apache OpenNLP POS Chunker: Dựa trên mô tả trong tài liệu của OpenNLP[1,2], chúng tôi thiết lập một tập tin chứa đựng những câu tiếng Việt theo cấu trúc sau:

- Trên mỗi dòng có ba cột được phân cách bởi một khoảng trắng. Nội dung của từng cột như sau: (i) Cột thứ nhất – một từ đơn hoặc một âm tiết của một từ ghép tiếng Việt; (ii) Cột thứ hai – nhãn tương ứng của từ đơn hoặc âm tiết; (ii) Cột thứ ba – một nhãn ngữ đoạn đặc biệt được gán cho từ đơn hay âm tiết này được tạo thành theo quy tắc: (a) gán "B-" hay "I-" vào trước tên nhãn ngữ đoạn mà từ đơn hay âm tiết này thuộc về, trong đó "B-" cho biết từ đơn hay âm tiết này là từ đầu tiên của ngữ đoạn, còn "I-" cho biết từ đơn hay âm tiết này là những từ tiếp theo của ngữ đoạn; (b) gán nhãn "O" nếu từ đơn hay âm tiết không thuộc về một ngữ đoạn nào.
- Hai câu được định dạng kế tiếp nhau thì được phân cách bởi một dòng để trống.

Xét cặp câu trong Ví dụ 2, các từ vựng và ngữ đoạn được gán nhãn trong tập tin huấn luyện như sau:

đổng	_HUMC	B-NP
ông	_HUMC	I-NP
hồi	_STPSE	B-VP
hộp	_STPSE	I-VP
mở	_TTOPE	B-VP
tủ	_NANIC	B-NP
.	.	O
ông	_PRCNH	B-NP
tái	_STMBM	B-PP
mặt	_NANIC	B-NP
đứng	_ITPCH	B-VP
vịn	_TTOPE	B-VP
thành	_NANIC	B-NP
tủ	_NANIC	I-NP
một	_CD	B-NP
lát	_TIMC	I-NP
.	.	O

Hình 6. *Định dạng huấn luyện gán nhãn ngữ đoạn cho cặp câu trong Ví dụ 2*

4.2. Chuyển đổi cấu trúc ngữ đoạn sang cấu trúc rút gọn

Nội dung chính của phần này là trình bày hai bước thực hiện chính trong pha thứ hai để chuyển đổi cấu trúc phân tích ngữ đoạn của từng câu về câu ở dạng rút gọn: (i) ánh xạ cấu trúc phân tích ngữ đoạn về dạng cấu trúc rút gọn; và (ii) tái tạo câu ở dạng rút gọn từ dạng cấu trúc rút gọn.

Trong những nghiên cứu trước ([44, 45, 46]) và nghiên cứu này, chúng tôi chấp nhận một cấu trúc rút gọn sẽ thuộc về một trong các dạng sau: (i) `S` $\rightarrow$ `NP + VP`; (ii) `S` $\rightarrow$ `NP + VP + NP`; (iii) `S` $\rightarrow$ `NP + PP`.

Để thực hiện bước thứ nhất trong pha này, ở nghiên cứu đầu tiên này chúng tôi thiết lập quy tắc trong việc lựa chọn ngữ đoạn phù hợp:

- **Quy tắc (1).** Nếu có nhiều `[NP]` liền nhau $\rightarrow$ Ưu tiên chọn lựa `[NP]` đầu tiên.
- **Quy tắc (2).** Nếu có nhiều `[VP]` liền nhau $\rightarrow$ Ưu tiên chọn lựa `[VP]` cuối cùng.
- **Quy tắc (3).** Nếu có nhiều `[PP]` liền nhau $\rightarrow$ Ưu tiên chọn lựa `[PP]` cuối cùng.
- **Quy tắc (4).** Nếu có nhiều cặp `[VP][NP]` liền nhau $\rightarrow$ Tách ra thành từng cặp ngữ đoạn để xử lý.
- **Quy tắc (5).** Nếu có nhiều cặp `[VP][NP]` cách nhau bởi dấu "," hoặc từ nối "và" $\rightarrow$ Tách ra thành từng cặp ngữ đoạn để xử lý.
- **Quy tắc (6).** Nếu có nhiều `[PP]` cách nhau bởi dấu "," hoặc từ nối "và" $\rightarrow$ Tách ra thành từng ngữ đoạn để xử lý.
- **Quy tắc (7).** Nếu có cặp `[PP][VP]` liền nhau $\rightarrow$ Tách ra thành từng ngữ đoạn để xử lý.

Tiền đề để thiết lập các quy tắc trên là dựa trên kinh nghiệm sử dụng tiếng Việt trong giao tiếp thông thường: (i) Khi có nhiều hành động liên tiếp được thực hiện, thông thường hành động cuối cùng là mục đích thực hiện chính của đối tượng chủ thể; (ii) Khi có nhiều đối tượng liên tiếp cùng giữ vai trò khách thể của một hành động, thông thường đối tượng đầu tiên là đối tượng được nhắm đến trực tiếp của hành động; (iii) Khi có nhiều cặp [hành động – đối tượng khách thể] liên tiếp cùng xuất hiện, những cặp này có vị trí tương đương trong ngữ cảnh xuất hiện. Dựa trên những quy tắc trên, chúng tôi đề xuất giải thuật tổng quát ban đầu để ánh xạ một cấu trúc phân tích ngữ đoạn về dạng rút gọn như sau:

Algorithm 1: Ánh xạ cấu trúc phân tích ngữ đoạn về dạng rút gọn.

Data: Danh sách các ngữ đoạn được gán nhãn của câu đầu vào.
Result: Danh sách gồm các phần tử đặc biệt, trong đó từng phần tử là một danh sách các ngữ đoạn được gán nhãn sau khi được rút gọn của một câu được tái tạo.

```
Khởi tạo danh sách OUT;
n = Số lượng ngữ đoạn ban đầu;
j = 1;
Khởi tạo A = "";
Duyệt ngữ đoạn ND trong danh sách;
if ND có dạng [PP] then
    {Xem Algorithm 2}
if ND có dạng [VP] then
    {Xem Algorithm 3}
if ND có dạng [NP] then
    {Xem Algorithm 4}
```

Algorithm 2: Xử lý với ND có dạng [PP].

```
Áp dụng Quy tắc (3), (6), (7);
if phù hợp Quy tắc (3) then
    Gán B = [PP] cuối cùng;
    Tạo danh sách DS(j) với: A và B;
    j = j + 1;
    Chuyển xem xét ngữ đoạn tiếp sau B;
if phù hợp Quy tắc (6) hoặc (7) then
    Gán B = ND;
    Tạo danh sách DS(j) với: A và B;
    j = j + 1;
    Chuyển xem xét ngữ đoạn tiếp sau B;
```

Algorithm 3: Xử lý với ND có dạng [VP].

```
Áp dụng Quy tắc (2), (4), (5);
if phù hợp Quy tắc (2) then
    Gán B = [VP] cuối cùng;
    if B là ngữ đoạn cuối trước khi hết câu hoặc gặp "," then
        Tạo danh sách DS(j) với: A và B;
    if ngữ đoạn sau B có dạng [NP] then
        Tạo danh sách DS(j) với: A và B, ngữ đoạn sau B;
    j = j + 1;
    Chuyển xem xét ngữ đoạn tiếp sau B;
if phù hợp Quy tắc (4) hoặc (5) then
    Gán B = [VP];
    Tạo danh sách DS(j) với: A và B, ngữ đoạn sau B;
    j = j + 1;
    Chuyển xem xét ngữ đoạn tiếp sau B;
if phù hợp Quy tắc (7) then
    Gán B = [VP];
    if B là ngữ đoạn cuối trước khi hết câu hoặc gặp "," then
        Tạo danh sách DS(j) với: A và B;
    if ngữ đoạn sau B có dạng [NP] then
        Tạo danh sách DS(j) với: A và B, ngữ đoạn sau B;
    j = j + 1;
    Chuyển xem xét ngữ đoạn tiếp sau B;
```

Algorithm 4: Xử lý với ND có dạng [NP].

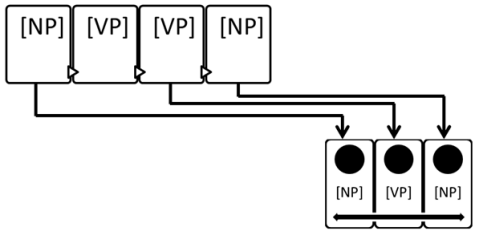
```
Áp dụng Quy tắc (1), (4), (5);
if phù hợp Quy tắc (1) then
    Gán A = [NP] đầu tiên;
if phù hợp Quy tắc (4) hoặc (5) then
    Chuyển xem xét ngữ đoạn [VP] trước đó;
```

Áp dụng Giải thuật 1 vào cấu trúc phân tích ngữ đoạn cho cặp câu trong Ví dụ 2, kết quả thực hiện chuyển đổi được minh họa như sau:

1) *Ánh xạ cấu trúc phân tích ngữ đoạn của câu thứ nhất về dạng rút gọn:*

$\Longrightarrow$ Diễn giải từng bước thực hiện Giải thuật 1 với các quy tắc bên trên:

- Bước 1: Xét ngữ đoạn đầu tiên:

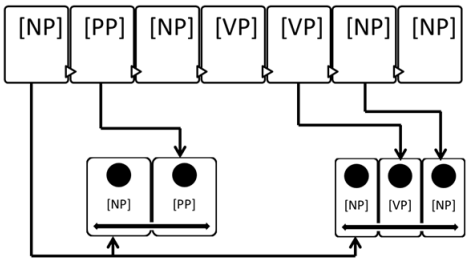


Hình 7. *Ánh xạ cấu trúc phân tích ngữ đoạn của câu "Đổng Ông hồi hộp mở tủ." về dạng rút gọn*

  - `A = [NP đổng__HUMC ông__HUMC ]`
- Bước 2: Xét cặp ngữ đoạn thứ hai và thứ ba:
  - `[VP hồi__STPSE hộp__STPSE ]` và `[VP mở_TTOPE ]`
  - Chọn lựa ngữ đoạn thứ ba `[VP mở_TTOPE ]` theo Quy tắc (2)
- Bước 3: Xét ngữ đoạn thứ tư:
  - `[NP tủ_NANIC ]`
  - Tạo danh sách DS(1) với: `A`, `[VP mở_TTOPE ]` và `[NP tủ_NANIC ]`

$\Longrightarrow$ Kết quả thứ nhất: `[NP đổng__HUMC ông__HUMC ] [VP mở_TTOPE ] [NP tủ_NANIC ]`

2) *Ánh xạ cấu trúc phân tích ngữ đoạn của câu thứ hai về dạng rút gọn:*



Hình 8. *Ánh xạ cấu trúc phân tích ngữ đoạn của câu "Ông tái mặt đứng vịn thành tủ một lát." về dạng rút gọn*

$\Longrightarrow$ Diễn giải từng bước thực hiện Giải thuật 1 với các quy tắc bên trên:

- Bước 1: Xét ngữ đoạn đầu tiên:
  - `A = [NP ông_PRCNH ]`
- Bước 2: Xét ngữ đoạn thứ hai:
  - `[PP tái_STMBM ]`
- Bước 3: Xét ngữ đoạn thứ ba:
  - `[NP mặt_NANIC ]`
  - Tạo danh sách DS(1) với: `A` và `[PP tái_STMBM ]`
- Bước 4: Xét cặp ngữ đoạn thứ tư và thứ năm:

  - [VP đứng_ITPCH] và [VP vịn_TTOPE]
  - Chọn lựa ngữ đoạn thứ năm [VP vịn_TTOPE] theo Quy tắc (2)
- Bước 5: Xét cặp ngữ đoạn thứ sáu và thứ bảy:
  - [NP thành__NANIC tủ__NANIC] và [NP một_CD lát_TIM]
  - Chọn lựa ngữ đoạn thứ sáu: [NP thành__NANIC tủ__NANIC]
  - Tạo danh sách DS(2) với: A, [VP vịn_TTOPE] và [NP thành__NANIC tủ__NANIC]

$\Longrightarrow$ Kết quả thứ hai: [NP ông_PRCNH] [PP tái_STMBM]; [NP ông_PRCNH] [VP vịn_TTOPE] [NP thành__NANIC tủ__NANIC]

Chúng tôi tiến hành tái tạo lại các câu ở dạng rút gọn của đoạn văn bản trong Ví dụ 2 dựa vào kết quả thứ nhất và kết quả thứ hai bên trên thành đoạn văn bản mới:

"*Đổng ông mở tủ. Ông tái. Ông vịn thành tủ.*"

Áp dụng những phương pháp và kỹ thuật trong các nghiên cứu trước ([44, 45, 46]), chúng tôi thực hiện hai bước xử lý đoạn văn bản mới này: (i) xác định đối tượng tiền ngữ cho đại từ "ông" ở câu thứ hai và câu thứ ba; (ii) biểu diễn ngữ nghĩa của đoạn văn bản dưới dạng cấu trúc DRS. Kết quả cuối cùng của phương pháp chuyển đổi cho đoạn văn bản trong Ví dụ 2 là cấu trúc DRS như sau:

```
[1, 2, 3]
đổng_ông(1,[đổng,ông],[object],[human])
tủ(2,[tủ],[object],[nonanimated])
mở(1,2,[mở],[action],[transitive])
tái(1,[tái],[state],[status])
thành_tủ(3,[thành,tủ],[object],[nonanimated])
vịn(1,3,[vị],[action],[transitive])
```

Hình 9. Cấu trúc DRS của đoạn văn bản "Đổng ông mở tủ. Ông tái. Ông vịn thành tủ."

5. Thử nghiệm

5.1. Tập hợp ngữ liệu thử nghiệm

Để thử nghiệm mô hình tổng quan giải pháp chuyển đổi đoạn văn bản tiếng Việt về dạng biểu diễn ngữ nghĩa và thực hiện tạo sinh tóm lược, chúng tôi tiến hành tập hợp những đoạn văn bản tiếng Việt bao gồm những câu có cấu trúc thông thường. Theo mục tiêu nghiên cứu của bài báo này, những đoạn văn bản tiếng Việt được tập hợp phải có sự xuất hiện của các đại từ hồi chỉ, thể hiện mối liên hệ giữa các câu. Các bước để thực hiện tổng hợp những đoạn văn bản tiếng Việt để thử nghiệm như sau:

- **Bước 1.** Xác định các đại từ hồi chỉ. Trong các nghiên cứu trước ([44, 45, 46]) và nghiên cứu này, chúng tôi tập trung xử lý các dạng đại từ hồi chỉ: (a) đại từ chỉ người ở ngôi thứ ba đứng một mình – "anh" / "cô" / "chị" / "ông" / "bà" / "bạn" / "em"; (b) những đại từ ở dạng (a) đứng cùng tính từ chỉ thị "ta" / "ấy" / "này"; (c) đại từ "nó".
- **Bước 2.** Tập hợp các đoạn văn bản tiếng Việt từ những nguồn sách giáo khoa Cấp một[3,4] cũng như sách song ngữ Anh – Việt[5,6] với những đặc điểm: (i) Từng đoạn văn bản bao gồm từ 2 đến 5 câu; (ii) Có sự xuất hiện của các đại từ hồi chỉ ở Bước 1; (iii) Từng câu có không quá 15 từ vựng (tính cả từ đơn và từ ghép); (iv) Từng câu thuộc dạng câu đơn, thể loại trần thuật.

Với hai bước trên, chúng tôi tập hợp ngữ liệu để tiến hành thử nghiệm như sau: (i) 1000 câu tiếng Việt dùng huấn luyện tách từ, gán nhãn từ vựng và gán nhãn ngữ đoạn với bộ công cụ OpenNLP[1,2]; (ii) 130 đoạn văn bản tiếng Việt gồm 400 câu làm ngữ liệu để tiến hành thử nghiệm phương pháp chuyển đổi và tóm lược.

5.2. Thiết kế thử nghiệm

Chúng tôi thực hiện thử nghiệm cho bộ ngữ liệu được tập hợp với hai nội dung:

1) Áp dụng phương pháp được trình bày trong bài báo để chuyển đổi từng đoạn văn bản về dạng rút gọn và thực hiện tạo sinh tóm lược từ dạng này theo các phương pháp được đề xuất trong [44, 45, 46].
2) Áp dụng hai phương pháp tóm lược được lựa chọn làm nền tảng để so sánh: (i) phương pháp của K. Filippova [21] xác định đường đi ngắn nhất trên đồ thị từ vựng để trộn những câu có quan hệ và tạo sinh câu rút gọn; (ii) phương pháp của F. Boudin và E. Morin [14] thực hiện cải tiến phương pháp của K. Filippova [21] với ý tưởng chính là đánh giá lại các câu kết quả trộn theo ngữ đoạn khóa. Hai phương pháp này được hiện thực trong mô-đun Takahe[7] với ngôn ngữ Python.

Ở bước tiền xử lý tách từ và gán nhãn với bộ công cụ OpenNLP[1,2], kết quả thu được dựa vào phương pháp đánh giá được trình bày trong hướng dẫn:

- Tỉ lệ đạt được trên 70% số câu đối với gán nhãn từ vựng. Chúng tôi thực hiện gán nhãn thủ công với những câu còn lại để tiếp tục thử nghiệm.
- Tỉ lệ đạt được trên 73% số câu đối với gán nhãn ngữ đoạn. Chúng tôi thực hiện gán nhãn thủ công với những câu còn lại để tiếp tục thử nghiệm.

[3]Sách giáo khoa Tiếng Việt lớp 1 tập 1, 2 do Đặng Thị Lanh chủ biên năm 2012. Nhà xuất bản Giáo dục Việt Nam. Bộ Giáo dục và Đào tạo.

[4]Sách giáo khoa Tiếng Việt lớp 2 – 3 – 4 – 5 tập 1, 2 do Nguyễn Minh Thuyết chủ biên năm 2012, 2014. Nhà xuất bản Giáo dục Việt Nam. Bộ Giáo dục và Đào tạo.

[5]Tập dịch những mẩu chuyện vui tiếng Anh – Mỹ tập 1 năm 1994. Nhà xuất bản Thành phố Hồ Chí Minh.

[6]Luyện đọc những mẩu chuyện tiếng Anh B năm 1994. Nhà xuất bản Thành phố Hồ Chí Minh.

[7]Takahe – a multi-sentence compression module – được phát triển bởi Florian Boudin cho ngôn ngữ Python 2 (tại https://github.com/boudinfl/takahe)

Tỉ lệ đạt được chưa cao bằng so với tỉ lệ thực thi gán nhãn từ vựng bằng vnTagger[8] là do một số yếu tố: (i) Bộ nhãn từ vựng khá chi tiết với những loại từ thuộc nhiều cấp; (ii) Số lượng ngữ liệu câu được huấn luyện chưa đủ nhiều.

Mô-đun Takahe[7] được thực hiện trong bộ công cụ lập trình NetBeansIDE[9] phiên bản 8.0.2 được tích hợp plugin python4netbeans8.0.2[10] dùng cho lập trình ngôn ngữ Python. Môi trường thực nghiệm được thực hiện trên hệ thống Linux Ubuntu phiên bản 12.04LTS 64bits được thiết lập sẵn phiên bản Python 2.7.3. Các bước thực hiện tóm lược các đoạn văn bản trong bộ ngữ liệu với hai phương pháp nền tảng như sau:

- **Bước 1.** Thực hiện gán nhãn từ vựng từng câu nguyên bản với bộ công cụ vnTagger[8]. Ở bước này, việc sử dụng bộ công cụ vnTagger[8] được thực hiện để so sánh hiệu quả tổng quát với phương pháp được trình bày trong bài báo. Tuy nhiên, mô-đun Takahe[7] được xây dựng phù hợp với bộ nhãn của dự án Penn Treebank [5] nên sau khi có kết quả gán nhãn từ vựng với bộ công cụ vnTagger[8], chúng tôi thực hiện thay thế thủ công nhãn hiện tại bởi nhãn tương ứng trong bộ nhãn của dự án Penn Treebank [5]. Với cách thức này, chúng tôi đạt được hai mục tiêu quan trọng: (i) tách từ và gán nhãn từ vựng cho các câu tiếng Việt (ii) bằng bộ nhãn của dự án Penn Treebank [5].

$\Longrightarrow$ Thực hiện gán nhãn từ vựng cho đoạn văn bản trong Ví dụ 2 với bộ công cụ vnTagger[8] với bộ nhãn đi kèm:

```
nonAR_vnTaggerTagset = ["Đổng/V ông/N hồi_hộp/V mở/V tủ/N ./.", "Ông/N
tái/A mặt/N đứng/V vịn/V thành/V tủ/N một/M lát/N ./."]
```

$\Longrightarrow$ Thực hiện thay thế thủ công với bộ nhãn của dự án Penn Treebank [5]:

```
nonAR_PTBTagset = ["Đổng/VB ông/NN hồi_hộp/VB mở/VB tủ/NN ./.", "Ông/NN
tái/JJ mặt/NN đứng/VB vịn/VB thành/VB tủ/NN một/CD lát/NN ./."]
```

- **Bước 2.** Thực hiện thủ công tiền xử lý đại từ hồi chỉ xuất hiện trong từng đoạn văn bản. Mục tiêu của việc tiền xử lý này là để so sánh khi thực hiện trộn những đoạn văn bản nhận được. Tiếp đó thực hiện như Bước 1 cho đoạn văn bản đã được tiền xử lý đại từ hồi chỉ.

$\Longrightarrow$ Tiền xử lý đại từ hồi chỉ cho đoạn văn bản trong Ví dụ 2, nhận được đoạn văn bản: "*Đổng ông hồi hộp mở tủ. Đổng ông tái mặt đứng vịn thành tủ một lát.*"

$\Longrightarrow$ Thực hiện gán nhãn từ vựng cho đoạn văn bản trong Ví dụ với bộ công cụ vnTagger[8] với bộ nhãn đi kèm:

```
AR_vnTaggerTagset = ["Đổng/V ông/N hồi_hộp/V mở/V tủ/N ./.", "Đổng/N ông/N
tái/JJ mặt/N đứng/V vịn/V thành/V tủ/N một/M lát/N ./."]
```

$\Longrightarrow$ Thực hiện thay thế thủ công với bộ nhãn của dự án Penn Treebank [5]:

[8]vnTagger – Vietnamese part-of-speech tagging (http://mim.hus.vnu.edu.vn/phuonglh/softwares/vnTagger)

[9]NetBeans IDE 8.0.2 (tại https://netbeans.org/)

[10]Python in NetBeans IDE 8.0.2 (tại http://plugins.netbeans.org/plugin/56795/python4netbeans802)

```
AR_PTBTagset = ["Đổng/VB ông/NN hồi_hộp/VB mở/VB tủ/NN ./.", "Đổng/NN ông/NN
tái/JJ mặt/NN đứng/VB vịn/VB thành/VB tủ/NN một/CD lát/NN ./."]
```

- **Bước 3.** Thực thi mô-đun Takahe[7] cho lần lượt `nonAR_PTBTagset` và `AR_PTBTagset`, nhận được các kết quả như sau:

$\Longrightarrow$ Kết quả thứ nhất. Thực thi trộn `nonAR_PTBTagset` với phương pháp của K. Filippova [21]. Kết quả nhận được là bốn câu rút gọn:

– "*Đổng ông hồi_hộp mở tủ .*"
– "*Ông tái mặt đứng vịn thành tủ .*"
– "*Đổng ông hồi_hộp mở tủ một lát .*"
– "*Ông tái mặt đứng vịn thành tủ một lát .*"

$\Longrightarrow$ Kết quả thứ hai. Thực thi trộn `nonAR_PTBTagset` với phương pháp của F. Boudin và E. Morin [14]. Kết quả nhận được là bốn câu rút gọn:

– "*Đổng ông hồi_hộp mở tủ một lát .*"
– "*Ông tái mặt đứng vịn thành tủ một lát .*"
– "*Đổng ông hồi_hộp mở tủ .*"
– "*Ông tái mặt đứng vịn thành tủ .*"

$\Longrightarrow$ Kết quả thứ ba. Thực thi trộn `AR_PTBTagset` với phương pháp của K. Filippova [21]. Kết quả nhận được là bốn câu rút gọn:

– "*Đổng ông hồi_hộp mở tủ .*"
– "*Đổng ông hồi_hộp mở tủ một lát .*"
– "*Đổng ông tái mặt đứng vịn thành tủ .*"
– "*Đổng ông tái mặt đứng vịn thành tủ một lát .*"

$\Longrightarrow$ Kết quả thứ tư. Thực thi trộn `AR_PTBTagset` với phương pháp của F. Boudin và E. Morin [14]. Kết quả nhận được là bốn câu rút gọn:

– "*Đổng ông tái mặt đứng vịn thành tủ một lát .*"
– "*Đổng ông hồi_hộp mở tủ một lát .*"
– "*Đổng ông tái mặt đứng vịn thành tủ .*"
– "*Đổng ông hồi_hộp mở tủ .*"

5.3. Đánh giá kết quả

Để đánh giá tự động kết quả tóm lược có được từ sự kết hợp phương pháp chuyển đổi được trình bày trong bài báo với những phương pháp tóm lược được đề xuất trong [14, 21], chúng tôi áp dụng độ đo ROUGE (C. Y. Lin [9, 10]). Công cụ được sử dụng là Rouge2.0_0.2[11] thực hiện tính toán các chỉ số F-score, Recall, Precision.

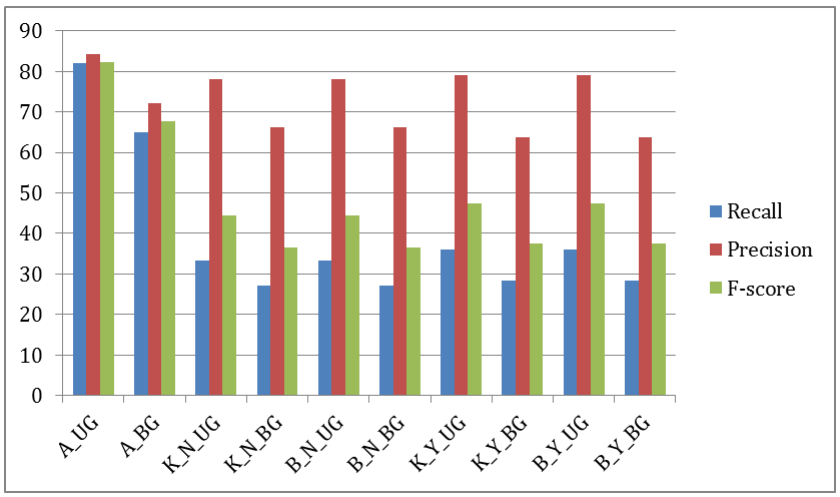
Trước tiên, chúng tôi thực hiện tập hợp những đoạn văn bản tóm lược thủ công cho từng đoạn văn bản thử nghiệm. Đối với từng đoạn văn bản thử nghiệm, số lượng đoạn

[11]ROUGE 2.0 - a Java Package for Evaluation of Summarization Tasks building on the Perl Implementation of ROUGE - được phát triển bởi Kavita Ganesan cho ngôn ngữ Java (tại http://www.rxnlp.com/rouge-2-0/)

văn bản tóm lược thủ công được tập hợp có thể dao động từ 3 đến 5. Danh sách những đoạn văn bản tóm lược thủ công này được sử dụng để so sánh và đánh giá chất lượng khi thực thi công cụ Rouge2.0_0.2[11] (được gọi là "reference summary").

Tiếp theo, chúng tôi lần lượt thực thi Rouge2.0_0.2[11] theo uni-gram và bi-gram cho những đoạn văn bản được tạo sinh tự động (được gọi là "system summary") từ các hệ thống lần lượt: (i) Phương pháp tóm lược kết hợp từ phương pháp chuyển đổi trong bài báo với các phương pháp được trình bày trong [44, 45, 46]; (ii) Phương pháp của K. Filippova [21] khi không thực hiện tiền xử lý đại từ hồi chỉ; (iii) Phương pháp của F. Boudin và E. Morin [14] khi không thực hiện tiền xử lý đại từ hồi chỉ; (iv) Phương pháp của K. Filippova [21] khi thực hiện tiền xử lý đại từ hồi chỉ; (v) Phương pháp của F. Boudin và E. Morin [14] khi thực hiện tiền xử lý đại từ hồi chỉ.

Kết quả thực hiện đánh giá bằng công cụ Rouge2.0_0.2[11] được thể hiện trong đồ thị ở hình 10 trong đó từng giá trị là trung bình các chỉ số Recall, Precision và F-score của tất cả các đoạn văn bản tóm lược thu được từ các phương pháp.



Hình 10. *Kết quả thực hiện so sánh và đánh giá với công cụ Rouge2.0_0.2[11]*

Những ký hiệu được sử dụng trong Hình 10: (i) A_UG – đánh giá phương pháp tổng quát của chúng tôi với uni-gram; (ii) A_BG – đánh giá phương pháp tổng quát của chúng tôi với bi- gram; (iii) K_N_UG – đánh giá phương pháp của K. Filippova [21] với uni-gram khi chưa tiền xử lý hồi chỉ; (iv) K_N_BG – đánh giá phương pháp của K. Filippova [21] với bi-gram khi chưa tiền xử lý hồi chỉ; (v) B_N_UG – đánh giá phương pháp của F. Boudin và E. Morin [14] với uni-gram khi chưa tiền xử lý hồi chỉ; (vi) B_N_BG – đánh giá phương pháp của F. Boudin và E. Morin [14] với bi-gram khi chưa tiền xử lý hồi chỉ; (vii) K_Y_UG – đánh giá phương pháp của K. Filippova [21] với uni-gram khi có tiền xử lý hồi chỉ; (viii) K_Y_BG – đánh giá phương pháp của K. Filippova [21] với bi-gram khi có tiền xử lý hồi chỉ; (ix) B_Y_UG – đánh giá phương pháp của F. Boudin và E. Morin [14] với uni-gram khi có tiền xử lý hồi chỉ; (x) B_Y_BG – đánh giá phương pháp của F. Boudin và E. Morin [14] với bi-gram khi

có tiền xử lý hồi chỉ.

Phân tích kết quả đạt được trong Hình 10, chúng tôi nhận thấy phương pháp tóm lược tổng quát của chúng tôi (bao gồm sự kết hợp phương pháp chuyển đổi được trình bày trong bài báo và những phương pháp tóm lược được đề xuất trong [44, 45, 46]) đạt được kết quả tốt so với một số phương pháp cơ sở khác được giới thiệu gần đây ([14, 21]).

Một điểm quan trọng được nhận thấy qua thử nghiệm là phương pháp chuyển đổi được trình bày trong bài báo cũng ảnh hưởng đến kết quả tóm lược cuối cùng. Phân tích sâu hơn, chúng tôi cũng ghi nhận một số điểm còn tồn tại trong phương pháp chuyển đổi sẽ được khắc phục trong những nghiên cứu tiếp theo:

- Các quy tắc được đặt ra trong Phần 4 chưa đủ để bao phủ các câu có thêm các ngữ đoạn mô tả trạng ngữ về thời gian, nơi chốn.

Ví dụ 3. "*Norah có căn nhà tranh trên bờ đá ở một vịnh lớn. Về mùa đông, nó rất dơ bẩn vì gió lớn và bụi biển.*"

$\Longrightarrow$ Câu thứ nhất và câu thứ hai đều có nội dung về thời gian và nơi chốn.

- Còn thiếu những quy tắc xử lý đối với những câu có cấu trúc dạng câu phức.

Ví dụ 4. "*Sáng hôm ấy, bé Thơ về, bông bằng lăng cuối cùng đã nở. Nhưng bông hoa lại nở cao hơn cửa sổ nên bé không nhìn thấy nó. Bé cứ ngỡ là mùa hoa đã qua.*"

$\Longrightarrow$ Câu thứ nhất và câu thứ hai có dạng câu phức.

$\Longrightarrow$ Câu thứ ba cũng có dạng cấu trúc phức tạp mà các quy tắc trong Phần 3 chưa xử lý được.

- Sự thiếu sót các quy tắc áp dụng cho những dạng câu có cấu trúc phức tạp hơn cũng dẫn đến sự chưa hoàn chỉnh của Giải thuật 1.

Trong nghiên cứu tiếp theo, chúng tôi sẽ áp dụng thêm những kiến thức về ngôn ngữ học để đề xuất các quy tắc phù hợp cũng như hoàn thiện Giải thuật ánh xạ.

6. Kết luận

Chúng tôi đã trình bày trong bài báo một phương pháp chuyển đổi một đoạn văn bản tiếng Việt nguyên bản bao gồm các câu có cấu trúc phức tạp sang một đoạn văn bản khác bao gồm các câu có cấu trúc đơn giản được định nghĩa trước và biểu diễn cấu trúc ngữ nghĩa của đoạn văn bản mới này. Phương pháp chuyển đổi bao gồm ba pha thực hiện chính: (i) Gán nhãn ngữ đoạn cho từng câu trong đoạn văn bản ban đầu; (ii) Ánh xạ từng câu được gán nhãn về các dạng câu rút gọn gồm những ngữ đoạn chính; (iii) Biểu diễn cấu trúc ngữ nghĩa của đoạn văn mới. Để thực hiện quá trình chuyển đổi, chúng tôi đã đề xuất việc định nghĩa một số nhãn từ vựng mới dựa trên sự phân loại câu theo nghĩa biểu hiện của lý thuyết ngôn ngữ học Ngữ pháp Chức năng ([17, 30]).

Bên cạnh đó, chúng tôi cũng đặt ra một số quy tắc chuyển đổi dựa trên kinh nghiệm thực tế sử dụng tiếng Việt trong giao tiếp thông thường.

Thử nghiệm cho thấy giải pháp chuyển đổi áp dụng được cho phần lớn các đoạn văn bản thử nghiệm. Tuy nhiên, chúng tôi cũng xác định một số điểm hạn chế hiện nay là: (1) Số lượng quy tắc còn chưa nhiều nên có những trường hợp việc rút gọn ngữ đoạn chưa chính xác; và (2) Chưa xem xét thêm những dạng câu ghép phức tạp hơn.

Trong nghiên cứu sắp tới, chúng tôi sẽ tiếp tục hoàn chỉnh bộ nhãn từ vựng và ngữ đoạn, đồng thời bổ sung thêm những quy tắc rút gọn và hoàn chỉnh giải thuật chuyển đổi.

Tài liệu tham khảo

[1] A. Berger, S.A.D. Pietra and V.J.D. Pietra, *"A Maximum Entropy Approach to Natural Language Processing"*, Computational Linguistics, vol. 22, no. 1, pp. 39-71, 1996.

[2] A. Khan and N. Salim, *"A Review on Abstractive Summarization Methods"*, Journal of Theoretical and Applied Information Technology, vol. 59, no.1, pp. 64–72, 2014.

[3] A. Ratnaparkhi, *"A maximum entropy model for part-of-speech tagging"*, Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1996.

[4] A. Ratnaparkhi, *A simple introduction to maximum entropy models for natural language processing*, Technical Report 97-08, Institute for Research in Cognitive Science, University of Pennsylvania, 1997.

[5] B. Santorini, *Part-of-speech Tagging Guidelines for the Penn Treebank Project*, Technical Report MS-CIS-90–47, Department of Computer and Information Science, University of Pennsylvania, 1990.

[6] C.F. Greenbacker, *"Towards a framework for abstractive summarization of multimodal documents"*, ACL HLT 2011, 75 (2011).

[7] C.S. Saranyamol and L. Sindhu, *"A Survey on Automatic Text Summarization"*, International Journal of Computer Science and Information Technologies, vol. 5, no. 6, pp. 7889–7893, 2014.

[8] C.S. Lee, Z.W. Jian and L.K. Huang, *"A Fuzzy Ontology and Its Application to News Summarization"*, IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 35, no. 5, pp. 859–880, 2005.

[9] C.Y. Lin, *"Looking for a Few Goods Metrics: ROUGE and its Evaluation"*, Proceedings of NTCIR Workshop 2004, Tokyo, Japan, 2004.

[10] C.Y. Lin, *"ROUGE: A Package for Automatic Evaluation of Summaries"*, Proceedings of the Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain, 2004.

[11] D. Das and A.F.T. Martins, *A Survey on Automatic Text Summarization*, Language Technologies Institute, Carnegie Mellon University, 2007.

[12] E. Krahmer, E. Marsi and P.V. Pelt, *"Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion"*, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics and Human Language Technologies, Columbus, Ohio, pp. 193–196, 2008.

[13] E. Lloret, *"Text Summarization: An Overview"*, paper supported by the Spanish Government under the project TEXT-MESS (TIN2006-15265-C06-01), 2008.

[14] F. Boudin and E. Morin, *"Keyphrase extraction for n-best reranking in multi-sentence compression"*, Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL- HLT 2013), Atlanta, Georgia, pp. 298–305, 2013.

[15] H. Kamp, *A Theory of Truth and Semantic Representation*, University of Amsterdam: Formal methods in the study of language, pp. 277–322, 1981.

[16] H.P. Le, T.M.H. Nguyen, M. Rossignol and A. Roussanaly, *"An empirical study of maximum entropy approach for part-of-speech tagging of Vietnamese texts"*, Proceedings of the Actes du Traitement Automatique des Langues Naturelles (TALN-2010), 2010.

[17] H.X. Cao, *Tiếng Việt: Sơ thảo ngữ pháp chức năng [Vietnamese: Brief of Functional Grammar]*, Nhà xuất bản giáo dục [Education Publisher], 2006.

[18] I.F. Moawad and M. Aref, *"Semantic graph reduction approach for abstractive Text Summarization"*, Proceedings of the 7th International Conference on Computer Engineering & Systems (ICCES), pp. 132-138, 2012.

[19] I. Mani and M.T. Maybury, *Advances In Automatic Text Summarization*, MIT Press, 1999.
[20] K.A. Ganesan, C.X. Zhai and J. Han, *"Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions"*, Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China, 2010, pp. 340–348.
[21] K. Filippova, *"Multi-Sentence Compression: Finding Shortest Paths in Word Graphs"*, Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China, pp. 322–330, 2010.
[22] K. Filippova and M. Strube, *"Dependency Tree Based Sentence Compression"*, Proceedings of the 5th International Natural Language Generation Conference, Salt Fork, Ohio, 2008.
[23] K. Filippova and M. Strube, *"Sentence Fusion via Dependency Graph Compression"*, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, 2008.
[24] K. Jezek and J. Steinberger, *Automatic Text summarization*, Vaclav Snasel (Ed.): Znalosti 2008, ISBN 978-80-227-2827-0, FIIT STU Brarislava, Ustav Informatiky a softveroveho inzinierstva, pp. 1–12, 2008.
[25] K.S. Jones, *"Automatic Summarizing: Factors and Directions"*, in: I. Mani and M. Marbury (Eds), Advances in Automatic Text Summarization, MIT Press, 1999.
[26] K.S. Jones, *Automatic Summarising: A Review and Discussion of The State of The Art*, Technical Report 679, Computer Laboratory, University of Cambridge, 2007.
[27] K. Toutanova and C.D. Manning, *"Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger"*, Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70, 2000.
[28] K. Toutanova, D. Klein, C. Manning and Y. Singer, *"Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network"*, Proceedings of HLT-NAACL 2003, pp. 252-259, 2003.
[29] M.A. Covington and N. Schmitz, *An Implementation of Discourse Representation Theory*, USA: Advanced Computational Methods Center, The University of Georgia, 1989.
[30] M.A.K. Halliday and C.M.I.M. Matthiessen, *An Introduction to Functional Grammar*, Third Edition, Hodder Arnold, 2004.
[31] M. Collins, *"Head-Driven Statistical Models for Natural Language Parsing"*, Computational Linguistics, vol. 29, no. 4, pp. 589–637, 2003.
[32] M. Collins and M.P. Marcus, *Head-driven statistical models for natural language parsing*, University of Pennsylvania, Philadelphia, PA, 1999.
[33] N.R. Kasture, N. Yargal, N.N. Singh, N. Kulkarni and V. Mathur, *"A Survey on Methods of Abstractive Text Summarization"*, International Journal for Research in Merging Science and Technology, vol. 1, iss. 6, pp. 53–57, 2014.
[34] O. Chaowalit and O. Sornil, *"An Automatic Approach to Generating Abstractive Summary for Thai Opinions"*, International Journal of Advancements in Computing Technology, vol. 6, no. 3, pp. 142–150, 2014.
[35] P. Blackburn and J. Bos, *Representation and Inference for Natural Language - Volume II: Working with Discourse Representation Structures*, Germany: Department of Computational Linguistics, University of Saarland, 1999.
[36] P.E. Genest and G. Lapalme, *"Framework for Abstractive Summarization using Text-to-Text Generation"*, Proceedings of the Workshop on Monolingual Text-to-Text Generation, Oregon, Portland, pp. 64–73, 2011.
[37] P.E. Genest and G. Lapalme, *"Text Generation for Abstractive Summarization"*, Proceedings of the 3rd Text Analysis Conference, Gaithersburg, Maryland, USA, 2010.
[38] P.E. Genest and G. Lapalme, *"Fully Abstractive Approach to Guided Summarization"*, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers – Volum 2, Jeju Island, Korea, pp. 354–358, 2012.
[39] R. Barzilay, K.R. Mckeown and M. Elhadad, *"Information fusion in the context of multi-document summarization"*, Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 550–557, 1999.
[40] R. Barzilay and K.R. McKeown, *"Sentence fusion for multidocument news summarization"*, Computational Linguistics, vol. 31, pp. 297–328, 2005.
[41] S. Gerani, Y. Mehdad, G. Carenini, T.NG. Raymond and B. Nejat, *"Abstractive Summarization of Product Reviews Using Discourse Structure"*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, pp. 1602–1613, 2014.
[42] S.K. Jagadish, K.G. Srinivasa and R.B. Eswara, *"A Comprehensive Analysis of Guided Abstractive Text Summarization"*, International Journal of Computer Science Issues, vol. 11, iss. 6, no. 1, pp. 115–121, 2014.

[43] S.M. Harabagiu and F. Lacatusu, *"Generating single and multi-document summaries with gistexter"*, Proceedings of Document Understanding Conferences, 2002.

[44] T. Tran and D.T. Nguyen, *"Modelling Consequence Relationships between Two Action, State or Process Vietnamese Sentences for Improving the Quality of New Meaning-Summarizing Sentence"*, International Journal of Pervasive Computing and Communications, vol. 11, no. 2, pp. 169–190, 2015. Emerald Group Publishing Limited. ISBN 1742-7371.

[45] T. Tran and D.T. Nguyen, *"Algorithm of Computing Verbal Relationships for Generating Vietnamese Paragraph of Summarization from The Logical Expression of Discourse Representation Structure"*, Vietnam Journal of Computer Science, vol. 3, iss. 1, pp. 35–46, 2016. (ISSN: 2196-8888 (Print) 2196-8896 (Online))

[46] T. Tran and D.T. Nguyen, *"Determining The Temporal Order between Two Vietnamese Process Sentences for Summarizing"*, Vietnam Journal on Information Technology and Communication – Research, Development and Application on Information and Communication Technology, no. 35, pp. 38–54, 2016. (ISSN: 1859-3534).

Ngày nhận bài 23-06-2016; Ngày chấp nhận đăng 18-01-2017.

■

Trần Trung Sinh năm 1985 tại Hải Dương. Tốt nghiệp ĐH ngành CNTT năm 2007 tại Trường ĐH Khoa học Tự nhiên, ĐH Quốc gia TP. HCM, Thạc sĩ chuyên ngành Khoa học máy tính năm 2012 tại Trường ĐH CNTT, ĐH Quốc gia TP. HCM. Làm Nghiên cứu sinh chuyên ngành Khoa học máy tính tại Trường ĐH CNTT, ĐH Quốc gia TP. HCM từ tháng 07/2012. Lĩnh vực nghiên cứu: Xử lý ngôn ngữ tự nhiên, Ngôn ngữ học máy tính.

Nguyễn Tuấn Đăng sinh năm 1972 tại Sài Gòn. Nhận bằng Cử nhân ngành Tin học tại Trường ĐH Mở Bán công TP. HCM năm 1996, Thạc sĩ ngành Tin học tại Viện Tin học sử dụng tiếng Pháp năm 2000, Thạc sĩ ngành Tin học tại Trường ĐH Khoa học Tự nhiên, ĐH Quốc gia TP. HCM năm 2003. Bảo vệ luận án Tiến sĩ ngành Tin học tại Trường ĐH Caen Basse-Normandie, Pháp năm 2006. Hiện là giảng viên tại Khoa Khoa học Máy tính, Trường ĐH CNTT, ĐH Quốc gia TP. Hồ Chí Minh. Chuyên ngành nghiên cứu: Xử lý ngôn ngữ tự nhiên, Ngôn ngữ học máy tính.