Enhancing Resume Content Extraction in Question Answering Systems through T5 Model Variants

Yuxin Luo[1,†], Feng Lu[2], Vaishali Pal[1] and David Graus[4]

[1]University of Amsterdam, Amsterdam, The Netherlands
[2]Randstad Groep Nederland, Diemen, The Netherlands
[3]University of Amsterdam, Amsterdam, The Netherlands
[4]Randstad, Diemen, The Netherlands

Abstract

Summarizing usable information from a large number of resumes is a tedious effort for all recruiters. The aim of this study is to explore the performance of the T5 model and its variants for automatic extraction of CV information by combining augmenting manual questions with a paraphraser under the same architecture, and fine-tuning a question and answering system using Dutch and English resumes in a multilingual version of the T5 model (mT5). Our results show that the quality of the generated answers varies considerably between information types, with superior performance for attributes such as basic information that rely on text extraction. However, there is more room for improvement in processing date-based information with multiple inputs, and inferring of multiple standardised answer choices.

Keywords

Natural Language Processing, QA system, Domain Information Extraction, Transformer models

1. Introduction

In the pre-selection phase of recruitment, recruiters reading CVs can be seen as a recruiter (internally) posing questions to a resume, and noting relevant information related to skills, work experience, and personalia. In this scenario, unstructured curriculum vitaes (CVs) could be treated as contextual background for machine learning-powered question-answering methods. In this paper, we explore the application of pre-trained large language models (LLMs) for Question Answering over CVs.

LLMs for QA typically learn to map questions and contexts to answers. In this paper, we rely on a dataset that consists of job seekers' CVs on one hand, and structured job seeker data on the other hand. With these two sources, that respectively represent the context and "answers" in the QA task, we have but one element missing to train an LLM for QA: the question, which typically is hand-engineered or based on templates [1].

One of the most popular and advanced models in the field of natural language processing is the Transformer which is a type of neural network architecture based on self-attention mechanisms. It typically solves a variety of NLP tasks by way of pre-training and fine-tuning. Pre-training refers to self-supervised training on large amounts of unlabelled data to generate a generalised language model. It learns to make inferences and predictions about missing or incorrect parts of the context by, for example, masking or replacing the input text sequence. The parameters and weights of the pre-trained model undergo fine-tuning through supervised training using downstream task data, which facilitates the process of transfer learning. The transformer approach is an advanced solution for several sub-tasks in NLP such as machine translation, text generation and so on, including question answering (QA) [2, 3, 4].

One central element in QA approaches is the pairing of questions to answers. A typical approach is to use templates, however using only templates may result in a low variety of questions and repetition of question types, or alternatively a huge amount of manual work for engineering a larger number of templates. In this paper, we find a middle ground and address the lack in diversity and variation from a small subset of hand-engineered templates, through employing transformer-based paraphrasing on a seed set of hand-engineered questions. After obtaining a sufficient number of questions for training, we apply a multilingual version of the T5 model (mT5) [5] to train our QA system.

This paper studies how this mT5 model can perform QA to extract information from different segments (*Basic information, Education, Skills*) of CVs. Our method follows two steps: First, we enrich our dataset that consists of (i) structured job seeker data, and (ii) unstructured job seeker CVs, by hand-engineering (iii) a small set of sample questions which we (iv) extend through transformer-

RecSys in HR'23: The 3rd Workshop on Recommender Systems for Human Resources, in conjunction with the 17th ACM Conference on Recommender Systems, September 18–22, 2023, Singapore, Singapore.
[†]Work done while on internship at Randstad Groep Nederland.
yuxin.luo@student.uva.nl (Y. Luo); feng.lu@randstadgroep.nl (F. Lu); v.pal@uva.nl (V. Pal); david.graus@randstadgroep.nl (D. Graus)

based paraphrasing. In doing so, we ensure a large and diverse enough set of <question, context, answer>-triples. Second, we use these expanded QA training sets for several downstream QA tasks.

In this paper, we aim to answer the following research question:

RQ1 To what extent can transformer-powered QA be used to meet the basic screening needs in a multilingual recruitment context?

We aim to answer this research question by answering the following sub questions:

RQ1.1 Can we apply transformers for paraphrasing manually generated questions, to increase the variety and volumes of training data for our QA models?

RQ1.2 How do transformer models deal with different CV segments with different types of answers? (e.g. basic information, skills stack, education, etc.)

RQ1.3 How accurate is the QA model in dealing with different languages of resumes?

The rest of paper is organized as follows: we discuss prior work in transformer-based Question Answering models, and CV-related datasets in Section 2. Next, in Section 3 we detail the methodology and overall experimental design. Then, Section 4 presents the results of our question paraphrasing approach, and downstream QA models separately. In Section 5 we reflect on the experimental results, and discuss the limitations of our experiments. Finally, in Section 6 we answer our research questions, and propose possible future research directions.

2. Related Work

In recent years, large pre-trained language models such as BERT [3], GPT-2 [6], and T5 [4] have achieved state-of-the-art results on many QA benchmarks. T5 employs a text-to-text approach, taking questions and contextual text as input and generating answers as output. It extends transfer learning boundaries by modeling human language use, understanding the importance of words in sentences. When comprehending text, attention is directed to specific words and their meanings. The novel innovation in T5 includes using a 'prefix' to specify the task, which is crucial for this study with two different downstream tasks. The tunability guaranteed by this feature is particularly important in the context of this study, where we aim to train two different downstream tasks.

2.1. Transformer-based Question-Answering

Transformer-based Question-Answering (QA) systems have achieved state-of-the-art performance in a wide range of domains, including open-domain QA, closed-domain QA, and factoid QA.

In open-domain QA, systems are required to answer questions from a broad range of topics. Here, transformer-based QA systems have shown to be effective in leveraging large-scale pre-training techniques and multi-task learning to improve their performance. For instance, models like T5, mT5, and ELECTRA [7] have shown to achieve high accuracy on the open-domain QA benchmark dataset SQuAD. For specific or well-documented domains such as wiki or medical data [8, 9], they have also demonstrated strong performance.

In addition, transfer learning has been applied to factoid QA, which involves answering questions that require a factual answer. For example, models like GPT-3 and XLNet have been applied to tasks like reading comprehension, summarization, and dialog generation, showing the ability to generate human-like responses and engage in natural language interactions.

Li et al. [10] constructed a CV-related database and trained it on a multi-turn question-answering system to obtain valid information. However, it was based on entity attribution relations and focused only on a limited number of four types of structured data (name, place of work, work duration, position), doing experiments on a dataset of under one thousand resumes. To the best of our knowledge, there is a few research work conducted in applying the QA systems in the human resource domain.

2.2. QA in specific domains

Applying transformers for QA in specific domains often faces the challenge of lacking large datasets of annotated question-answer pairs. Existing solutions to acquire such annotated datasets include crowdsourcing, textbook problem repositories, transfer learning, and domain-specific ontologies.

Crowdsourcing practices are often used, i.e. manual annotation of data with a large workforce, which usually results in a high volume of training data in a short period of time and a high accuracy rate. However, successful crowd-sourcing is often run by large companies at a significant cost and its superiority is currently only proven in consumer products [11]. In the field we are applying it to, the need for information from the perspective of the professional recruiter may not match that of outside volunteers. And there is no question banks in place that can meet the demand.

Transfer learning is pre-trained on a large dataset (open source corpus), and once the model has a better

understanding of the language, it no longer needs to be heavily annotated with questions and then put into a domain-specific dataset after fine-tuning [8, 9]. In the other way, problems are generated in an ontological structure by identifying corresponding concepts and relationships through entities in the learning domain. That is, conducting a graph-based question generation or rule-based question generation [12, 13].

Both approaches have domain and coverage limitations. Re-matching question-answer pairs after question generation is required for QA system input and the data used in this experiment is bilingual. To address these limitations, we apply data augmentation to obtain training questions, including paraphrased questions generated using a transformer model after key-value interrogation for structured data, ensuring question-answer pair matching for each input question.

3. Methodology

In this section, we introduce each step of our QA approach. Initially, we preprocess the original dataset, organizing attributes into three subsets. Next, we create manual questions and apply the template approach to establish baseline results. Finally, we proceed with question augmentation as the upstream task for T5 architecture. After evaluating the paraphrased questions quality, the QA system is trained in the downstream and the results are evaluated with our baselines.

3.1. Data

Our proprietary dataset contains two sets of job seeker data, that can be mapped to each other: first, we have parsed CVs, i.e., textual data extracted from CV source files (e.g., PDF or DOC) through the Amazon Textract service [14]. It should be noted that these texts are available in English and Dutch.

Next, we have structured data, either provided by the job seekers themselves upon registration or submitted by recruiters. The structured data includes basic information such as name, address, contact details, but also education experience, in addition to skills.

In this paper, we focus on three different information types: basic information, education, and skills. Table 3.1 shows the features in each module and the key attribute to link them (candidate_id).

First, **basic information** is mostly of textual nature (i.e., string type data), and spans personal identifiable information (PII) data such as name, address, and contact information such as email address.

The second information type, **education data**, contains both textual (string) data (education level description), and datetime data (education start and end dates). As shown in the table, the education data represents the level of completed education (as chosen from a list of five education levels), and not e.g., the name of educational institute or program. Educational levels are numbered from one to five.

Finally, structured **skills data** is provided in a similar format. In the process of providing structured data on skills, job seekers select the appropriate description from a multiple choice box containing 270 options to demonstrate the skills they possess.

3.2. Baseline QA approaches

Microsoft's Presidio[1] serves as the initial baseline for our experiment, primarily designed for detection of personally identifiable information (PII) in text. In this study, we employ Presidio to identify the basic information in each CV, namely 'PERSON' for names, 'DATE_TIME' for birth dates, 'EMAIL_ADDRESS' for email addresses, 'LOCATION' for addresses, and 'PHONE_NUMBER' for mobile phone numbers. It is essential to note that Presidio may detect multiple entities for a single attribute. To ensure consistency, we concatenate all detected entities into a single string-format entity, which serves as the final extracted entity for comparison with the target text.

Presidio does not support extracting education and skills data out of the box. For this reason, we introduce a keyword-based textual segmentation method as the second baseline in the experiment. In this approach, we segment CV texts into different sections based on keywords, and use the underlying sections as extracted answers. We created a dictionary with section names as keys, and six keywords lists (three in English and three in Dutch) as values. For example, for the attribute *mobile*, we have keywords *telephone, mobile, phone* in English and keywords *telefoonnummer, mobiel, contactgegevens* in Dutch.

We perform a search for these keywords over the parsed CV texts, marking their starting positions. Once a first keyword is found, the subsequent keyword is marked as the first section's boundary. Using these keyword positions, we segment the CV text into different parts between consecutive keywords. The text between two keywords represents the first keyword's corresponding section's content. For specific attributes, we used the extracted section text as the answer. It's worth noting that to enable concise segmentation, we included some sections that did not cover the specific attributes we focused on for information extraction (e.g., the "work" keyword to segment work experiences).

[1] https://microsoft.github.io/presidio/

Table 1
Overview of the features in datasets

Data type	Attibutes	Description	Data type	Occuring values
-	candidate id	Public key for all entries	Integer	-
CV	text	Parsed text in resume	String	-
	language	Language used in this resume	String	English, Dutch
Basic	name	Name of the owner of the resume	String	-
	email	Email address of the candidate	String	-
	mobile	Phone number of the candidate	String	-
	address	Residence address of the candidate	String	-
	birth date	Date of birth of the candidate	Datetime	YYYY-MM-DD
Education	level	Desciption of this education input	String	-
	level code	The education level code corresponding to this education	Integer	1-5
	start date	The date of the start of the educational experience described in the entry	Datetime	YYYY-MM-DD
	end date	The date of the end of the educational experience described in the entry	Datetime	YYYY-MM-DD
Skill	OMS	Description of this skill input	String	-
	skill id	The skill id corresponding to this skill	Integer	1-274

3.3. Data augmentation for question reformulation

For training QA approaches, larger numbers of question and answer pairs typiucally yield better results, even if there is a saturation effect [15], which occurs at around 1,000 samples for transfer learning according to Agrawal et al. [16]. Therefore, we generate additional questions by applying transformers for rephrasing manually written questions.

The three primary information types, namely basic information, education and skills, have different attributes. Basic information comprises five attributes (*name, email, mobile, birth_date, address*), education consists of three attributes (*level, start_date, end_date*), and the skills module includes a single attribute (*skill*).

Attribute values are taken from the structured data, and we apply template-based question generation for getting the questions per attribute. Ten manual questions are initially generated for each attribute, followed by changing formulations from first to third person to double the number to twenty, adding diversity (e.g., *'What is your name?'* becomes *'What is the name of the candidate?'*). Manual questions undergo review by academic and human resource industry experts.

We paraphrase each question using a linguistic rephrasing framework called Parrot [17], which is built on the fine-tuned paraphraser model in the T5 architecture, and is supposed to be efficient for data augmentation [18].

For each of the twenty original questions per attribute, Parrot paraphrases are generated with the 'diversity' option and an 'adequacy' setting of 0.9, ensuring semantic similarity. We deduplicate (paraphrased and original) questions, and repeat the process three times, yielding an average of 1,371 questions per attribute.

3.4. mT5 model

We utilize the PyTorch implementation from Hugging Face for fine-tuning, initially employing the pre-trained mT5 small model. Our training process involves a triplet of data, namely questions, answers, and contextual text (parsed CV text).

The contextual text and questions are input sequences to the mT5 model, generating implicit representations. The answers serve as the target sequence for training.

Special tokens are added to input sequences, and we explore two different prompting methods:

- **A** The question and context are combined and passed to the model. Special tokens, <CLS> (start), <SEP> (split), and another <SEP> (end), are used to separate the inputs.
- **B** Context and question are encoded with distinct prefixes. A <RESUME> token marks the start of the context, followed by the CV text, and <QUESTION> tokens denote the question's beginning.

In addition, for contexts, we add addtional prefixes that indicate their languages: <nl> for Dutch, and <en> for English. The token <SEP> is also used to divide two questions when multiple questions are input, which only happens for education information, where next to level, we aim to extract start and end dates.

Example:
Context: 'Name: Mike, age: 30, gender: male'
Question: 'What is your name? '
Input A: <en> <CLS> Name: Mike, age: 30, gender: male <SEP> What is your name? <SEP>
Input B: <en> <RESUME> Name: Mike, age: 30, gender: male <QUESTION> What is your name? <SEP>

For the background input (i.e., CV text) we need to take into account mT5's limit on the maximum length of the

input text, which is 2,048 tokens. The average length of our CV text is around four 400 tokens, with seventy-five percent of the data being less than 573 tokens. But there are extreme data points with more than 10,000 tokens.

For Basic information and Education information, which we found to usually appear at the beginning of the CV, we use a maximum input limit of 512 tokens, i.e., we use only the first 512 tokens and truncate the remaining.

For the Skills module, it is difficult to judge where they are more likely to appear in the CV, so in this case we use a maximum input limit of 1,000 tokens after which we truncate.

3.5. Experimental setup

The dataset is divided into a training set, a validation set and a test set in a ratio of 6:2:2. We choose the Adafactor as optimizer under setting suggested by Shazeer and Stern [19].

Considering the computational costs of resources for three separate QA models (trained per information type), unless otherwise stated, each single QA model training-validation-testing process will be performed on a random set of 20,000 inputs.

3.6. Evaluation

As the question dataset is derived from our our annotations, the evaluation is divided into two parts. First, we evaluate the question paraphraser, and next, we evaluate the quality of the answers generated by the QA system.

3.6.1. Question paraphraser evaluation

The expectation for the generated questions is that they use a diversity of forms (linguistically different) compared to the original questions in terms of wording and grammar, but that they are identical in terms of meaning, in the sense that they correspond to the answers to the original questions (semantic agreement). In order to verify that the generated questions meet the requirements, we use BLEU [20] (BiLingual Evaluation Understudy), METEOR [21] (Metric for Evaluation of Translation with Explicit ORdering), and BERTScore [22] to evaluate the generated questions from a linguistic and semantic perspective respectively.

BLEU measures sentence similarity via word overlap between generated and reference sentences. Scores range from 0 to 1, with higher scores indicating greater word usage similarity. We use n-grams of size 1 and 2 due to short question lengths (5-15 words).

METEOR assesses sentence similarity by considering word matching, word order correctness, and word sense matching, capturing semantic similarity and relatedness.

BERTScore relies on a pre-trained BERT language model, using word embeddings to calculate sentence similarity. It emphasizes semantic relevance over lexical overlap and selects the BERT-base model based on evaluation requirements.

The ideal solution would exhibit low BLEU, and high METEOR and BERTScore, which indicates that the generated questions differ in syntax and surface form from the standard manual questions, but remain semantically similar.

In addition to these performance metrics, we also calculate the number of questions augmented, and the vocabulary size difference between the manual and generated questions, i.e., *nQuestions* indicates the number of generated questions, and *nWords* indicates the number of unique words added to the question set after the paraphrasing for each purpose.

3.6.2. QA evaluation

The three main evaluation metrics, BLEU, METEOR and BERTScore, are also used to evaluate the quality of answers in the QA system in both lexical and semantic perspectives. The base model of the BERTScore is the BERT multilingual base model (cased).

Moreover, there are specific metrics for each information type. We define EM (Exact Match), for which the basic information should be most sensitive to, as there is only one standard and specific answer for each single question.

In the Education module case, we need to take into account that a CV may contain information of multiple educational experiences from different periods, and hence it may take multiple questions at once. When assessing this, in addition to three metrics above, we define PM (Partial Match) to evaluate each separate answer's quality; we distinguish PM^{level}, which represents the level of the education is extracted exactly, PM^{start}, which indicates the start date of this education entry is answered correctly, and PM^{end}, which denotes the response to the end date.

In the skill information, the attention is focused on the METEOR score as the structured skill answer is provided in phrase format, which means that there's a possibility that it is not in the resume text but come out with the candidate's former experience or minds.

4. Results

Table 2 shows the evaluation of questions generation by comparing generated question with the manual questions for each attributes. Table 3 illustrates the evaluation results of question answering quality by comparing the generated answers with the structured data as target answers for each module. It includes approach (*Method*),

Table 2
Results of evaluation of generated questions.

Type	Attributes	BLEU-1	BLEU-2	METEOR	BERTScore	nWords	nQuestions
	All	46.51	35.25	59.75	53.99	12155	12335
Basic	Name	46.40	35.60	59.83	49.48	223	1135
	Email	46.40	35.41	60.74	55.70	174	1598
	Mobile	49.31	38.25	61.22	55.27	174	1316
	Address	45.68	33.47	56.18	52.53	249	1460
	Birth date	44.48	32.48	53.86	53.99	162	**931**
Education	Level	43.95	31.25	56.51	51.42	214	1581
	Start date	49.41	40.40	**66.70**	**59.05**	116	1314
	End date	49.44	40.11	65.07	58.26	176	1585
Skills	Skill	**43.27**	**29.82**	55.96	49.83	253	1415

input format (*Input*), training configuration (batch size (*ba*), and number of epochs (*ep*)), lexical and semantic matching quality of answers (*BLEU, METEOR, BERTScore, EM, PM*).

4.1. Question paraphraser

The results under attribute *All* represent all questions as a whole, regardless of information type or attribute, and compares and evaluates the rephrased question set with the original question set.

The overall quality of the generated questions presented in the Row 1 of the table 2 suggests relatively low similarity to the manual questions in terms of words and syntax, with BLEU scores below 50, while exhibiting high semantic similarity.

Specifically, we observed results for the following evaluation metrics; we find an inevitable positive correlation between lexical overlap and semantic similarity, with Skills having the lowest BLEU score, and also bearing the only BERTScore under 50. We find that several other attributes (e.g., Mobile, Start date, End date) exhibit slightly higher word overlap (i.e., BLEU scores) than others. For the number of questions generated (nQuestions), we find that the number of Birth attribute questions is just below 1,000, despite taking exactly the same settings and steps as the other attributes, which yield roughly between 1,100 and 1,600 additional questions. In addition, for each attribute, we find that the act of expanding the question sets by paraphrasing, substantially increases the vocabulary of the questions (nWords).

4.2. Question Answering

In this section, we proceed to describe the results of our methods per information type.

4.2.1. Basic information

For the basic information part, we evaluated the performance of three methods: rule-based, Presidio, and our finetuned model, using different evaluation metrics: BLEU, METEOR, BERTScore, and Exact Match (EM).

First, we turn to top rows in Table 3 for the answer extraction performance comparison between the different methods .

Starting with the rule-based method, it achieved modest results across the evaluation metrics. It attained low BLEU and METEOR scores of 15.71%, 9.26% and 28.45%. The BERTScore of 69.67% is acceptable, and an Exact Match (EM) score of 19.6%. Although the rule-based method served as a foundational baseline, its performance revealed limitations in capturing the intricacies of language and extracting information with high precision.

The Presidio method, on the other hand, exhibited noticeable improvements over the rule-based technique. Especially, it garnered higher BLEU scores of 36.99% and 27.31%, and 31.4% of the answers were given accurately (EM). Presidio's ability to incorporate context and contextually aware patterns enabled it to surpass the rule-based method in all respects.

However, it is important to note that despite the improved performance of Presidio compared to the rule-based approach, the fine-tuned model with suitable configuration outperformed both of them in all metrics. In particular, the best-performing results in the fourth row are a BLEU-1 score of 90.64 and a BERTScore of a remarkable 98.38 percent. These scores were significantly higher than those obtained by both the rule-based method and Presidio, indicating the clear superiority of the transformer approach in text processing and information extraction tasks.

Rows three to ten in the table 3 show how the trans-

Table 3
The results of generated answers. *ba, ep* refer to batch, epoch respectively

Information type	Method	input	ba	ep	BLEU-1	BLEU-2	METEOR	BERTScore	EM	PM^{level}	PM^{start}	PM^{end}
Basic	Rule-based	-	-	-	15.71	9.26	28.45	69.67	19.6	-	-	-
	Presidio	-	-	-	36.99	27.31	42.95	75.24	31.4	-	-	-
	mT5	A	4	1	3.1	0.65	2.27	32.28	0.0	-	-	-
	mT5	A	4	2	**90.64**	**55.37**	**74.9**	**98.38**	**87.02**	-	-	-
	mT5	A	4	4	88.51	54.28	73.52	97.8	85.35	-	-	-
	mT5	A	8	1	34.12	28.33	32.87	80.63	30.42	-	-	-
	mT5	A	8	2	80.19	47.79	66.65	95.32	74.05	-	-	-
	mT5	B	4	1	3.78	1.37	2.88	31.72	0.0	-	-	-
	mT5	B	4	2	86.04	51.5	71.44	97.53	78.05	-	-	-
	mT5	B	8	1	12.74	8.06	11.01	65.11	8.57	-	-	-
Education	Rule-based	-	-	-	17.99	8.89	23.92	67.87	-	34.98	-	-
	mT5	A	4	1	53.13	34.09	47.33	89.83	-	48.72	6.02	7.8
	mT5	A	4	2	**58.82**	**42.0**	**54.34**	91.19	-	54.12	**29.82**	**32.7**
	mT5	A	8	1	58.1	41.06	53.19	91.09	-	53.9	28.78	29.9
	mT5	B	4	2	58.14	40.67	51.99	**91.41**	-	**55.45**	29.02	29.12
Skills	Rule-based	-	-	-	1.63	0.03	1.75	57.99 -	-	-	-	
	mT5	A	4	1	12.38	6.19	12.71	70.58	-	-	-	-
	mT5	B	4	1	2.84	1.59	5.33	64.3	-	-	-	-
	mT5	B	4	4	14.02	8.78	15.31	70.63	-	-	-	-
Basic_en	mT5	A	4	2	85.71	57.7	72.89	97.1	78.26	-	-	-
Basic_nl	mT5	A	4	2	90.98	55.21	75.02	98.43	**87.62**	-	-	-
Edu_en	mT5	A	4	2	50.56	31.93	45.63	88.79	-	41.39	31.34	32.54
Edu_nl	mT5	A	4	2	59.78	43.18	55.35	91.48	-	55.61	29.65	32.72

former model behaves with different hyperparameter settings. For our mT5 model, the best input format is input A (with *<CLS><SEP>* token encoding), which generally performs better than input B for the same training settings. By fine-tuning the training configuration of the model and allocating computational resources, we were able to achieve optimal answer quality in all scores by setting the batch size and number of epochs to four and two, respectively, with over 90% overlap of individual words and 87.02% accuracy in exact answer matching.

In summary, while Presidio showed an improvement over the rule-based method, the transformer method with the trained hyperparameters demonstrated the best performance overall, making it the most suitable choice for answering questions about the basic information.

4.2.2. Education

In this education module analysis, we compared two distinct methods: the traditional rule-based approach and the modern transformer method, using BLEU, METEOR, BERTScore, and Partial Match (PM) evaluation metrics. In the first row of this module, The rule-based method yielded results with low BLEU scores of 17.99 and 8.89, a METEOR score of 23.92, a BERTScore of 67.87, as it demonstrates similar capabilities in the basic information module. PM^{level} indicates that the answer obtained correctly matches the level of education currently entered. The rule-based method offered a foundation for text processing to achieve accuracy at 34.87%. However, the main idea of the rule-based approach lies in segmenting the CVs and obtaining a small whole snippet of educational information without dedicated rules for linking the time and education level within it, so no corresponding matching scores about time can be derived.

For the transformer experiments of the education questions, the model configuration of four batches and two epochs still outperformed others. However, with education, the different input formats resulted in smaller differences than with the basic information QA model. Input B (*<RESUME><QUESTION>* token encoding) only performed slightly better than input A when we look at the first partial match, meaning that input B may have helped to better capture information on the level of education in the CV. For either input method and model setting, the accuracy of answers for education level (PM^{level}) was significantly higher than for start date (PM^{start}) and end date (PM^{end}). Moreover, in the training of this module, the increase in batch size was to some extent detrimental to the scores of each response quality.

4.2.3. Skills

In the skill module, we evaluated the performance of two methods: the rule-based approach and the advanced transformer method, utilizing key evaluation metrics: BLEU, METEOR, and BERTScore.

Both methods, unfortunately, fell short of our expectations in terms of overall performance. The rule-based approach has a much lower efficiency in extracting skill information than the above two modules. BLEU-1 and METEOR scores drop to single digits, 1.63 and 1.75 respec-

tively. The transformer method, while showcasing better results, still left room for improvement, with scores of BLEU-1 12.38%, METEOR 12.71%, and BERTScore 70.58%.

For the fine-tuning of the transformer model in the skills section, input A (*<CLS><SEP>* token encoding) continued to show its strengths, although the overall lexical matching scores were not as good as the two modules above, i.e., the task is harder. When training the model, this module took much longer to run on the same amount of data.

4.2.4. Multilingual performance

There were differences in the quality and accuracy of answer generation between languages, with Dutch CVs scoring slightly higher than English in both the basic information (with 90.98 vs. 85.71 BLEU-1 respectively) and education modules (with 59.78 vs. 50.56 BLEU-1 respectively) under the same model. For the three attributes in the education module, however, the model showed homogeneity across languages for the questions related to education start and end dates.

As we mentioned in Section 3, all prefix tokens and questions were in English. However, the overall quality of the responses is still slightly better in Dutch than in English, with the Dutch CV even scoring 9.36% higher than the English CV on the measure of exact match in basic module.

5. Discussion

In this section, we discuss question augmentation first, and then the performance comparison between the mT5 transformer model, and our two baselines: Presidio for basic information, and the rule-based method, following by the different features of transformer model across the modules and language.

5.1. Question augmentation usage

Table 2 shows the result of the comparison between manual and paraphrased questions. The expanded data generated by the paraphraser is able to meet the needs of downstream question and answer system training. Manual inspection revealed that the paraphrased questions vary, but largely maintain the meaning of the input question. The relatively large number of rephrased questions is due to them in some cases not being a complete change of questioning style to the original question, but a simple modification of the word abbreviation or tone. In other cases, the question departs completely from the input question in textual overlap (e.g., *"how should I call you?"* given as input: *"what's the candidate's name?"*).

As the rephrased questions have low similarity in word syntax to the manual questions, it can be expected that the data will be useful in increasing the diversity and coverage of the training sample. At the same time, the high semantic similarity of the rephrased questions to the human questions will help to improve the performance of the QA system in terms of semantic understanding and answering questions.

See an example of a paraphrased question in Table 4 in different similarity, given as input question: *What's the candidate's name?*. With this example, we can more intuitively feel that only high semantic similarity paraphrasing can satisfy the downstream training needs, i.e., the left column of the table. At the same time, low linguistic similarity, paired with high semantic similarity, will effectively diversify the question set, thus ensuring the robustness of our Q&A system.

5.2. Overview of transformer models performance

We used a sufficient number (over 1,000 for each attribute) of questions to fine-tune the mT5 model. Initially, we expected transformer models to outperform baseline methods due to their ability to learn complex patterns in textual data. However, experimental results varied. Tweaking model settings and input forms significantly improved the answer quality scores for basic information. Under the 4-batch and 1-epoch configuration, transformer models performed worse than baseline methods, possibly due to limited exposure to diverse CV data. Increasing the batch size improved transformer model performance, allowing for more efficient parallel processing during training. Increasing the number of epochs to 2 with a batch size of 4 had a positive impact, enabling the models to refine their learned representations and capture finer patterns in resume data. This resulted in improved the exact match accuracy to 87% under input 1 format (*<CLS><SEP>* token encoding).

Interestingly, when comparing the performance of models trained with 4 batches and 2 epochs to those trained with 4 batches and 4 epochs, the former performed better. This suggests that after a certain point, increasing the number of epochs may not yield significant performance improvements and may even lead to over-fitting on the training data in basic information module.

For both the education and skills information types, the underlying transformer models went beyond the rule-based baseline. The different input forms and the configuration of model training all had no significant effect on the acquisition of educational information.

Under the skills module, input A performs significantly better than input B.

Furthermore, the potential of the models that we have fine-tuned may be wider than the evaluation results suggest. For example, a generated answer to a question about

Table 4
Example of paraphrased questions given as input: *What's the candidate's name?*. Please note these rephrased questions are illustrational: due to our adequacy threshold at 0.9, the right column's low semantically similar rephrased questions are discarded.

	High semantic similarity	Low semantic similarity
High linguistic similarity	What is the candidate's name?	What's the candidate's age?
Low linguistic similarity	**tell me the name of the candidate**	tell me the email of the candidate

address is *Heilige Geeststraat*, while the target answer (i.e., ground truth) is *Eilige Geeststraat*, which means the generated answer does not yield an exact match. In fact, we found that there is no street named *Eilige Geeststraat* but the street *Heilige Geeststraat* is the correct street name. In certain instances, individuals may include only partial address information in their CVs. Through the utilization of the fine-tuned mT5 model, it becomes evident that this technology adeptly facilitates the precise extraction of said partial address information.

The original CV mentions the existent one, which means there is a possibility that the ground truth provided by the candidate or filled by the recruiter are incorrect due to, e.g., a typo, but our fine-tuned mT5 QA model demonstrates the ability to extract the correct information from the original text.

5.3. Modules and Language-specific Adaptations

Our experiments focused on a question and answer system for three CV information types: Basic information, Education, and Skills. The transformer model excelled at obtaining Basic information, demonstrating high accuracy and strong lexical and semantic similarity to original answers, likely due to direct extraction without complex reasoning. For Education information, the model's accuracy varied; it performed better for education level questions than for date-related queries. Date data's various formats contributed to this discrepancy. Obtaining educational information involved both transforming date data and mapping educational descriptions to levels, with CVs containing multiple instances of educational information. The model's self-attentive mechanism might not fully capture the sequential nature of numeric data, as it was originally designed for natural language text. The uniform date format in structured data differed from real CVs, where dates are often imprecise, leading to a large amount of data ending on the first or last day of the month. The Education module achieved high semantic similarity but low lexical similarity to original answers, using different expressions to understand and respond to questions rather than directly copying phrases from the original text [23].

The CV's Skills information posed the most significant challenge in comparing different information types vertically.

Semantic and lexical similarity scores between model-generated and target answers diverge widely, despite having relatively similar semantics and low lexical overlap. The diversity in skill descriptions likely contributes to this outcome. During structured data entry, candidates choose skill descriptions from 270 options with different granularities, e.g., "Microsoft Word" or "Office Suite" being skills that partly overlap, which leads to variations in how similar skills are expressed.

Personal preferences for customizing descriptions also pose a challenge. The low word overlap score in the segmentation method suggests a wide range of structured options, allowing room for personal preferences. Candidates select descriptions that match their experience and understanding, which may not align with the model's training data. As a result, vocabulary similarity in generated answers varies significantly.

In such cases, the model must select the most appropriate skill description based on context and question, requiring additional reasoning and comprehension.

The performance of the fine-tuned model in terms of answer generation for both languages showed little difference in semantic similarity scores in the same module, with Dutch slightly outperforming English. However, for text-based question answer accuracy, basic information and education level, the model outperformed English in terms of extracting and reasoning about information in Dutch (9.36% and 14.22% higher respectively). This has to be attributed to the data itself, where the proportion of raw data in the dataset is 90% for Dutch, and around 10% for English; and where the description of education level is a common expression under the Dutch education system. This results in our model being better at capturing the textual information in Dutch CVs.

5.4. Limitations

Despite the results achieved in this study in the task of extracting CV information in the form of questions and answers, there are still some limitations that need to be considered.

Firstly, there was a data imbalance in the training data in terms of language, as described above, where our dataset had an over-representation of Dutch CVs. This may have some impact on the model's performance in the case of English CVs or when English skill information is included. The dominance of structured information in Dutch phrases in the skills module may lead to difficulties and reduced performance of the model when dealing with English skill descriptions.

Secondly, annotation limitations also need to be considered. The annotation process for the skills module involves candidates selecting appropriate descriptions through multiple choice boxes, and this subjective nature of the labelling process may lead to differences in personal preferences and expressions. Inconsistencies in the labelling criteria may affect the lexical overlap and semantic similarity scores of the answers generated by the model.

6. Conclusion

In this study, we explored an approach combining upstream and downstream tasks, augmented by a paraphraser of the T5 model for manual questions, and fine-tuned using the mT5 model. We experimented with the CV information extraction task in question-and-answer format for three different information types, and evaluated the performance of the model. Based on the results and discussions we obtained, we are able to provide answers to the research questions.

6.1. RQ1.1: Transformers for dual tasks

The questions generated by the paraphraser under the T5 architecture are sufficient in number, have low overlap with the manual question vocabulary and are semantically similar. Thus the need for fine-tuning the question and answer system for the mT5 model can be met. By tuning the hyperparameter settings and input forms of the training model, we found that for intercepted answers, the tuned transformer model was able to exploit its ability to learn complex textual contexts and thus far outperform the keyword-segmentation approach. In the hyperparameter setting, increasing the number of epochs trained outperformed a larger batch of models, for the same conditions. Labeling the context-question-ends with the same token helps the model to understand the context better.

6.2. RQ1.2: Information types

Through comparison and analysis between the different information types, we found that the fine-tuned model performed superior in question answering of basic information, with high accuracy and high semantic similarity to the original answers.

However, for education information, the model was more accurate for education level-related questions than for education start or end date-related questions, and scored high semantic similarity but relatively low lexical similarity. This may be due to the fact that the acquisition of educational information involves reasoning about educational descriptions (institution names or degrees with levels) and the diversity of date data in formats may pose a challenge to the model.

In addition, in skills information, the semantic similarity scores and lexical similarity scores of the model-generated answers differed significantly from the target answers, with similar semantics but low lexical overlap, and the overall performance was not as good as the first two modules. This may be due to the subjective nature of the annotation process, resulting in diversity and lexical variation in the generated answers.

6.3. RQ1.3: Cross-lingual performance

In terms of language, the model maintained a relatively good semantic similarity, although the fact that there was a data imbalance in our dataset prevented us from drawing absolute conclusions and the accuracy of the model's extractive answers on English CVs decreased.

6.4. Future work

The possibilities for future work are varied. One is multilingual support, where future research could expand the dataset to include more samples in multiple languages to improve the adaptability and generalisation of the model to more linguistic contexts.

Another direction is to address the performance differences between different modules in the CV information extraction task. For information extraction on skills, consider trying to manipulate their terms as mentioned by Smith et al. [24] or drawing on external references (Wikipedia, LinkedIn) as Kivimäki et al. [25] did to make skill annotation standards consistent.

The exploration of utility is also worth looking at. The data we have used is a generic CV dataset, but practical applications may face domain-specific or firm-specific needs. Future research could therefore explore domain adaptive techniques to enable models to be adapted to the needs of different domains or companies.

References

[1] H. Hussein, M. Elmogy, S. Guirguis, Automatic english question generation system based on template driven scheme, International Journal of Computer Science Issues (IJCSI) 11 (2014) 45.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach (2019). arXiv:1907.11692.

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv preprint arXiv:1910.10683 (2019).

[5] L. Xue, X. Sun, F. Yang, Z. Dai, H. Zhang, H. Fang, Y. Guo, J. Li, X. Li, M. Liu, et al., mt5: A massively multilingual pre-trained text-to-text transformer, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 45–51.

[6] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, URL https://openai.com/blog/language-unsupervised/ (2018).

[7] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555 (2020).

[8] S. Soni, K. Roberts, Evaluation of dataset selection for pre-training and fine-tuning transformer language models for clinical question answering, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020, pp. 5532–5538. URL: https://aclanthology.org/2020.lrec-1.679/.

[9] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, X. Lu, Pubmedqa: A dataset for biomedical research question answering, arXiv preprint arXiv:1909.06146 (2019).

[10] X. Li, F. Yin, Z. Sun, X. Li, A. Yuan, D. Chai, M. Zhou, J. Li, Entity-relation extraction as multi-turn question answering, arXiv preprint arXiv:1905.05529 (2019).

[11] H. Simula, The rise and fall of crowdsourcing?, in: 2013 46th Hawaii International Conference on System Sciences, 2013, pp. 2783–2791. doi:10.1109/HICSS.2013.537.

[12] G. Lecorvé, M. Veyret, Q. Brabant, L. M. Rojas Barahona, Sparql-to-text question generation for knowledge-based conversational applications, in: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, 2022, pp. 131–147.

[13] K. Han, T. Castro Ferreira, C. Gardent, Generating questions from wikidata triples, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association (ELRA), 2022, pp. 277–290.

[14] Amazon Web Services, Amazon textract, https://aws.amazon.com/textract/, n.d. Accessed: April 20, 2023.

[15] R. Puri, R. Spring, M. Patwary, M. Shoeybi, B. Catanzaro, Training question answering models from synthetic data, 2020. URL: https://arxiv.org/abs/2002.09599. doi:10.48550/ARXIV.2002.09599.

[16] P. Agrawal, C. Alberti, F. Huot, J. Maynez, J. Ma, S. Ruder, K. Ganchev, D. Das, M. Lapata, Qameleon: Multilingual qa with only 5 examples, 2022. URL: https://arxiv.org/abs/2211.08264. doi:10.48550/ARXIV.2211.08264.

[17] P. Damodaran, Parrot: Paraphrase generation for nlu., 2021.

[18] L. F. A. O. Pellicer, T. M. Ferreira, A. H. R. Costa, Data augmentation techniques in natural language processing, Applied Soft Computing 132 (2023) 109803.

[19] N. Shazeer, M. Stern, Adafactor: Adaptive learning rates with sublinear memory cost, in: International Conference on Machine Learning, PMLR, 2018, pp. 4596–4604.

[20] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[21] S. Banerjee, A. Lavie, Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, in: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005, pp. 65–72.

[22] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, arXiv preprint arXiv:1904.09675 (2019).

[23] E. Wallace, Y. Wang, S. Li, S. Singh, M. Gardner, Do NLP models know numbers? probing numeracy in embeddings, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-

IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5307–5315. URL: https://aclanthology.org/D19-1534. doi:10.18653/v1/D19-1534.

[24] E. Smith, A. Weiler, M. Braschler, Skill extraction for domain-specific text retrieval in a job-matching platform, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings 12, Springer, 2021, pp. 116–128.

[25] I. Kivimäki, A. Panchenko, A. Dessy, D. Verdegem, P. Francq, H. Bersini, M. Saerens, A graph-based approach to skill extraction from text, in: Proceedings of TextGraphs-8 graph-based methods for natural language processing, 2013, pp. 79–87.