Journal of Applied Science, Engineering, Technology, and Education *ISSN: 2685-0591 (online)*
Vol. 7 No. 1 (2025)
https://doi.org/10.35877/454RI.asci4209

Pre-Trained Transformer-Based Approach for Arabic Question Answering: A Comparative Study

Amani Jamal* & Kholoud Alsubhi

Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia

Abstract

Question answering (QA) is one of the most challenging yet widely investigated problems in Natural Language Processing (NLP). Question-answering (QA) systems try to produce answers for given questions. These answers can be generated from unstructured or structured text. Hence, QA is considered an important research area that can be used in evaluating text understanding systems. A large volume of QA studies was devoted to the English language, investigating the most advanced techniques and achieving state-of-the-art results. However, research efforts in the Arabic questionanswering progress at a considerably slower pace due to the scarcity of research efforts in Arabic QA and the lack of large benchmark datasets. Recently many pre-trained language models provided high performance in many Arabic NLP problems. In this work, we evaluate the state-of-the-art pre-trained transformers models for Arabic QA using four reading comprehension datasets which are Arabic-SQuAD), ARCD, AQAD, and TyDiQA-GoldP datasets. We fine-tuned and compared the performance of the AraBERTv2-base model, AraBERTv0.2-large model, and AraELECTRA model. In the last, we provide an analysis to understand and interpret the low-performance results obtained by some models.

Keywords: Arabic Question Answering, Arabic Pre-trained Language Models, Reading Comprehension

Received: 10 March 2025 Revised: 17 April 2025 Accepted: 25 April 2025

1. Introduction

Question answering (QA) is an interactive human-computer process expressed in a natural language query. Question answering systems offer a solution to achieve the exact answer to a question written in natural language. There are two main approaches for QA systems: text-based QA and knowledge-based QA. QA system based on the knowledge base finds the most related answer-sentence to the question from the structured knowledge base and then returns the exact answer to the user. Text-based question answering problems are often formulated as reading comprehension problems where the task is to extract the answer from a given context passage. In these problems, the performance of the question-answering system is highly influenced by the size and quality of reading comprehension datasets. Thus, the progress of Arabic reading comprehension is still behind compared to the English language due to the lack of large and high-quality reading comprehension datasets in the Arabic language.

Question answering research has received considerable attention in recent years due to the importance of QA applications. Therefore, QA systems can be classified in several ways based on different factors, such as domain coverage, document retrieval approach, and answer extracting technique (Biltawi, Tedmori, & Awajan, 2021). Domain coverage can be open or closed. The open-domain QA systems retrieve the answers from the Web, and closed-domain QA systems retrieve the answer from specific documents. The document retrieval process can be done using rule-based methods or different search techniques. For answer extracting techniques, several approaches were examined, including rule-based, machine learning, or deep learning. Unlike the rule-based approaches that require feature engineering techniques and hands crafted rules, deep learning models require minimum knowledge about the lexicon and the syntax of the language. Most recently, deep learning models and, more specifically, transformer-based models have been the most effective and widely adopted approach for many natural language tasks. Instead of the sequence word dependency

* Corresponding author.
E-mail address: atjamal@kau.edu.sa

architecture of recurrent neural network models, transformer-based models process textual information in parallel and apply self-attention mechanisms to compute attention weights that estimate the influence of each word on another. Since 2018 several pre-trained models

were released, such as the bidirectional encoder representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2019) that contributed heavily to the success of many NLP applications. Similar to other NLP tasks, pre-trained transformer-based models, more specifically BERT-based models, have been successfully utilized in many QA systems. Several studies have examined pre-trained transformed-based model in English QA; however, few studies investigated the effects of using pre-trained models in the Arabic QA tasks despite the availability of several Arabic pre-trained transformer models.

Arabic pre-trained transformer-based models like AraBERTv2-base, AraBERTv0.2-large, and AraELECTRA [4, 2] have been successfully adopted in tasks like classification, named entity recognition, and question answering providing the state-of-the-art performance on many Arabic NLP tasks [4, 5].

In this study, we investigate the performance of three Arabic pre-trained transformer-based models, which are AraBERTv2-base, AraBERTv0.2-large, and AraELECTRA with different datasets on the task of question answering.

To create our QA system, we, in particular, performed the following: First, fine-tune the Arabic pre-trained transformerbased models AraBERTv2-base, AraBERTv0.2-large, and AraELECTRA separately on four widely used QA datasets. Second, we combined all datasets and used them to fine-tune the models to analyze the effect of utilizing large datasets on the performance of the models. We compare the performances of the three fine-tuned models using four annotated question answering datasets for reading comprehension task which are Arabic-SQuAD, ARCD (Mozannar, El Hajal, Maamary, & Hajj, 2019), AQAD (Atef, Mattar, Sherif, Elrefai, & Torki, 2020), and TyDiQA-GoldP (Clark et al., 2020).

The main contribution of this paper is to present a comparative study of pre-train transformer-based models in Arabic QA and analyze factors, like size and quality of datasets, that could affect the system's performance. Furthermore, we contribute to improving the performance of the QA system by optimizing the hyper-parameters such as learning rate and number of epochs.

This paper is organized as follows. Section 2 presents works related to Arabic transformer-based QA. In Section 3 and 4, we present details of the models and datasets used in this study. Experiments and evaluation of the system are presented in Section 5. Results and discussion are presented in Section 6.

2. Related Works

Compared to the English language question answering, the Arabic question answering systems progress very slow due to the shortage of natural language processing resources and the datasets of Arabic question answering. Therefore, most QA research in Arabic attempts to create question answering systems using information retrieval-centric techniques by using a set of rules to choose the answer and ranking paragraphs sentences and named entities that are considered answers. In this section, we will focus on studies that used deep learning techniques and transformers models in the Arabic QA reading comprehension tasks.

To compensate for the lack of large reading comprehension datasets several researchers translated available English QA datasets into Arabic. Mozannar et al. (Mozannar, El Hajal, Maamary, & Hajj, 2019) translated English SQuAD 1.1 QA dataset (Rajpurkar, Zhang, Lopyrev, & Liang, 2016)to Arabic and used and fine-tuned the transformer model Multilingual BERT(mBERT) for the QA task. They also developed the Arabic Reading Comprehension Dataset (ARCD) composed of 1,395 questions. The ARCD dataset experiment with BERT-based reader achieved a 50.10 F1-score, and the experiment on the Arabic-SQuAD dataset achieved 48.6 F1-score (Mozannar, El Hajal, Maamary, & Hajj, 2019).

Antoun et al. (Antoun, Baly, & Hajj, 2020a) pre-trained BERT (Devlin, Chang, Lee, & Toutanova, 2019) specifically for the Arabic language named AraBERT. They fine-tuned the model on the Question Answering (QA) task to select a span of text that contains the answers for a given question. The results showed that the performance of AraBERT compared to multilingual BERT from Google is much better, and this model achieved state-of-the-art performance on most Arabic NLP tasks (Antoun, Baly, & Hajj, 2020a).

Also, Atef et al. (Atef, Mattar, Sherif, Elrefai, & Torki, 2020) presented the Arabic Question-Answer dataset (AQAD), a new Arabic reading comprehension dataset. This dataset consisting of more than 17,000 questions and Arabic

Wikipedia articles matched the articles used in the well-known SQuAD dataset (Rajpurkar, Zhang, Lopyrev, & Liang, 2016). They fine-tuned BERT-Base un-normalized cased multilingual model(mBERT), which is trained on multiple languages, including Arabic. They also evaluated the BiDAF model and used Arabic fastText embedding. They achieved 33 Exact Match(EM) and 37 F1-score and 32 EM and 32 F1-score using mBert and BiDAF model, respectively.

Antoun et al. (Antoun, Baly, & Hajj, 2020b) develop the pre-training text discriminators for Arabic language understanding named ARAELECTRA. The discriminator network has the same architecture and layers as a BERT model. So in fine-tuning approach, they added a linear classification layer on top of ELECTRAs output and fine-tuned the whole model with the added layer on reading comprehension tasks. They evaluate the model on many Arabic NLP tasks, including reading comprehension. The question answering task measures the models reading comprehension and language understanding capabilities.

They used the Arabic Reading Comprehension Dataset(ARCD) (Mozannar, El Hajal, Maamary, & Hajj, 2019) and the Typologically Diverse Question Answering dataset (TyDiQA) (Clark et al., 2020). This model achieved state-of-the-art performance in Arabic QA by obtaining 71.22 F1-score on ARCD dataset and 86.68 F1-score on TyDiQA dataset.

Clark et al. (Clark et al., 2020) presented a question answering dataset covering 11 typologically diverse languages called TyDiQA. The dataset covered three question answering tasks; passage selection task, minimal answer span task, and gold passage task. The Gold Passage (GoldP) task is similar to current reading comprehension datasets. In the gold passage task, only the gold answer passage is given instead of using the entire Wikipedia article. They fine-tuned mBERT jointly on all languages of the TyDiQA gold passage. The result on the Arabic language reached 81.7 F1-score.

3. Models

In NLP, every text is tokenized and encoded to numerical vectors to be fed into the models. To encode the text, there are many representations like Bag of Words (BOW) (Mikolov, Grave, Bojanowski, Puhrsch, & Joulin, 2018), Term Frequency-Inverse Document Frequency (TF-IDF) (Hiemstra, 2000), Word2Vec (Church, 2017), and FastText (Mikolov, Grave, Bojanowski, Puhrsch, & Joulin, 2018). Bag of Words representation built a vocabulary from all the unique words and assigned to a one encoded vector; however, it faces problems when the vocabulary size grows (Zhang, Jin, & Zhou, 2010). The TF-IDF is usually used in information retrieval systems by determining the relevance of the word to a document in a set of documents; this model depends on overlaps between the query and the document. This model has a limitation on learning more difficult representations (Mishra & Vishwakarma, 2015). Word2Vec creates distributional vectors to capture the semantic so the similar meaning words will have similar vectors (Church, 2017). FastText learns the word representation that relies on the skip-gram model from Word2Vec. FastText cannot handle the word that has different possible meanings based on its context. So, it is not suitable for contextualized representations in question answering tasks (Liao, Shi, Bai, Wang, & Yao, 2017). On the other hand, all high performing models used pre-trained transformer-based (e.g., BERT) models to produce contextualized representations of words (Lan et al., 2020).

Transfer learning in NLP is performed by fine-tuning pre-trained language models for different tasks with a small number of examples producing improvement over other traditional deep learning and machine learning. This approach utilizes the language models that had been pre-trained in a self-supervised manner. It is helpful for us to use transfer learning models pre-trained on large multilingual corpus like Wikipedia. From that inspiration, we have chosen an Arabic version of the original BERT the AraBERTv2-base, AraBERTv0.2-large (Antoun, Baly, & Hajj, 2020a), and AraELECTRA (Antoun, Baly, & Hajj, 2020b), as our selected models for the Arabic question-answering task. We have implemented three pre-trained models on the following public release dataset: ARCD, Arabic-Squad, TyDiQA-GoldP, and AQAD. We worked in PyTorch and used Huggingfaces Pytorch implementation. The models' components are detailed in the following sections.

3.1. BERT

Bidirectional Encoder Representation from Transformers (BERT) is designed to pre-train deep bidirectional representations from unlabeled texts by jointly conditioning on both left and right context in all layers. BERT processes the words in relation to all the other words in a sentence rather than processing them sequentially. This helps pick up features of a language by looking at contexts before and after an individual word. BERT achieves state-of-the-art results on multiple natural language processing tasks, including the Stanford Question Answering Dataset (SQuAD v1.1 and SQuAD v2.0) question answering (Devlin, Chang, Lee, & Toutanova, 2019).

3.1.1. BERT architecture

In general, the transformer architecture uses the encoder-decoder method, an encoder for reading the input sequence, and the decoder generates predictions for the task. The transformer encoder simultaneously reads the whole input sequence, unlike the sequence to sequence models, which reads the left-to-right or right-to-left sequence. BERT architecture is based on implementation described in (Vaswani et al., 2017), which uses self-attention to study contextual relations between words. The architecture of the BERT base includes 12 transformer blocks, 768 hidden layers, and 12 attention heads, a total of around 110M parameters. BERT large architecture contains 24 transformer blocks, 1024 hidden layers, and 16 attention blocks with a staggering number of 340M parameters. The input of BERT is a sequence of tokens that are mapped to embedding vectors. The output of the encoder is a sequence of vectors indexed to tokens.

3.1.2. Input output representation

The input representation is able to represent both a single and a pair of sentences (e.g., Question, Answer) in a single token sequence. A "sequence" refers to the input token, which may be a one sentence or two sentences together.They used WordPiece embeddings (Wu et al., 2016) with a 30k vocabulary. The initial token of the sequence is a unique token called the [CLS] token. If a couple of sentences are provided as input, then the sentences are separated using a separator token [SEP]. Segment embedding is used to designate whether a sentence is from sentence 1 or 2. The final input is the summation of the token embedding and segment embedding.

3.1.3. Pretraining BERT

BERT is pre-trained using two unsupervised tasks called masked language modeling and next sentence prediction. For the masked language modeling, the deep bidirectional representation of the input BERT masks some of the input tokens and then predicts those masked tokens during the training stage. The final output from the encoder is delivered into a softmax layer to get the word predictions. The next sentence prediction is used to capture sentence level relations which are helpful in question answering tasks because the question answering tasks are based on learning the relationship between two sentences (Devlin, Chang, Lee, & Toutanova, 2019).

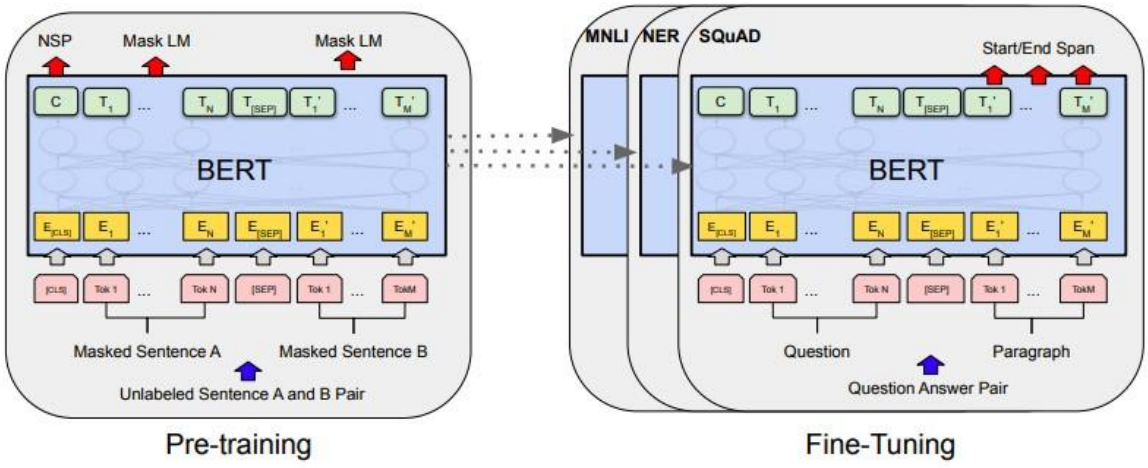

Fig. 1. Pre-training and fine-tuning procedures for BERT (Devlin, Chang, Lee, & Toutanova, 2019)

3.1.4. Fine-tuning BERT

BERT model was fine-tuned in many NLP tasks, including the question-answering task. For a given question and the passage that contains the answer. The question is assigned the segment 'A' embedding, and the passage is given the sentence 'B' embedding. BERT needs to highlight the "span" of text that could be the correct answer. This is computed as the dot product of the possibility between which token marks are the start of the answer and which token marks are the end. After taking the dot product between the output embeddings and the AŸ start A˘Z weights, the softmax applied´ to produce a probability distribution over all words. The word with the highest probability of being the start token is the one that chooses (McCormick, 2019). Figure 1 shows pre-training and fine-tuning procedures for BERT.

3.1.5. AraBERT

AraBERT is a representation model built for Arabic NLP tasks. AraBERT architecture is based on the BERT model, a stacked Bidirectional Transformer Encoder (Devlin, Chang, Lee, & Toutanova, 2019). AraBERT base is made of 12 encoder blocks, 768 hidden layers, 12 attention heads, 512 maximum sequence length, and a total of 110M parameters. AraBERT large has 24-layer, 1024 hidden dimension, 16 attention heads,and 336*M* parameters. AraBERT contains additional preprocessing before the models pre-training to better fit the Arabic language.The models' are available in different versions AraBERTv0.1&v1base, AraBERTv0.2&v2-base, and AraBERTv0.2&v2-large. The performance of this model achieved state-of-the-art in most Arabic NLP tasks like Sentiment Analysis (SA), Named Entity Recognition (NER), and Question Answering (QA) (Antoun, Baly, & Hajj, 2020a). Fine-tuning aims to start with a pre-trained model and then train it on the custom dataset's raw text. The masked LM task will be used to fine-tune the language model. We fine-tune the AraBERT model on our datasets and adjust model hyper-parameters like epochs, learning rate, batch size, and optimizer. We turn data for training, validation, and testing into internal BERT representation. Fit technique on the learner object is used to start the model training.

The method accepts the following parameters: epoch, learning rate, and optimizer schedule type. Make a new learner object and run the fit method to repeat the experiment with other parameters.

3.2. ELECTRA

ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) is a new and more efficient self-supervised language representation learning approach. ELECTRA, similar to the Generative adversarial network (GAN) (Goodfellow et al., 2020), trains two transformer models, the generator and discriminator. The model, in particular, performs a pre-training task called Replaced Token Detection (RTD) to replace some tokens with plausible alternatives sampled from a small generator model. In this way, the discriminator model tries to predict whether a token is an original or a replacement by a generator sample instead of training a model to predicts the identities of the masked tokens (See Fig 2. They were initially releasing three pre-trained models, which are small, base, and large. ELECTRA achieves state-of-the-art results on the SQuAD 2.0 dataset in 2019 (Clark, Luong, Le, & Manning, 2020).

AraELECTRA is an Arabic language representation model pre-trained using the RTD (Antoun, Baly, & Hajj, 2020b) methodology on large Arabic text corpora.AraELECTRA consists of 12 encoder layers, 12 attention heads, 768 hidden size and 512 maximum input sequence length for a total of 136M parameters. Figure 2 shows the replaced token detection pre-training task for AraELECTRA.The model was fine-tuned by adding a linear classification layer on top of ELECTRA's output and adding a layer on QA tasks to the discriminator network, which has the same structure and layers as a BERT model.

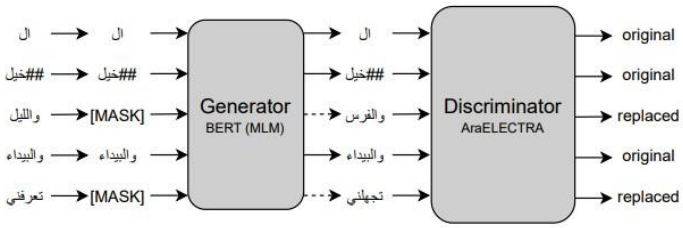

Fig. 2. Replaced Token Detection pre-training approch (Antoun, Baly, & Hajj, 2020b)

4. Dataset

We used four public release datasets for the reading comprehension task. The format of the four datasets matches the well-known SQuAD 1.0. dataset (Rajpurkar, Zhang, Lopyrev, & Liang, 2016).

Each example in the dataset has a context paragraph, question, and answers, structured as follows:

1. Context: The paragraph from which the question is asked
2. Qas: The list includes the following:
   (a) Id: A unique id for the question
   (b) Question: A question.
   (c) Answers: A list that includes the following:

(i) answer: the answer to the question
(ii) answer-start: contain the starting index of the answer in the context.

4.1. Arabic-SQuAD

Arabic translated version of SQuAD (Stanford Question Answering Dataset) (Rajpurkar, Zhang, Lopyrev, & Liang, 2016), a reading comprehension dataset composed of 48,344 annotated questions on a selection of Wikipedia articles. SQuAD contains 48,344 questions on 10,364 paragraphs for 231 articles (Mozannar, El Hajal, Maamary, & Hajj, 2019).

4.2. Arabic Reading Comprehension Dataset (ARCD)

ARCD was created by Mozannar et al. (Mozannar, El Hajal, Maamary, & Hajj, 2019) in 2019 and contained 1,395 questions posed by crowdworkers on Arabic

Wikipedia articles. This dataset was written by proficient Arabic speakers. The Arabic Reading Comprehension Dataset (ARCD) was used in multiple QA systems and showed a good performance.

4.3. TyDiQA

TyDiQA is a Multilingual human-annotated question-answer dataset including 11 typologically diverse languages with

204K question-answer pairs. The data is collected directly in each language without the use of translation and written without seeing the answer. The dataset was designed for the training and evaluation of automatic question answering systems. The size of the Arabic dataset is 15,645 question-answer pairs. The primary tasks of this dataset are the Passage selection task (SelectP) and Minimal answer span task (MinSpan). Also, the secondary task is the Gold passage task (GoldP) which means given a passage that contains the answer, predicts the single contiguous span of letters that answers the question. In this research, we used the Arabic TyDiQA-GoldP dataset (Clark et al., 2020).

4.4. AQAD

AQAD is an Arabic questions and answering dataset consists of 17,911 questions, extracted from 3,381 paragraphs, collected from 299 Arabic Wikipedia articles that match the one used in the SQuAD dataset. AQAD is a largesized dataset collected automatically without crowdworkers. AQAD did not apply machine translation for the paragraphs used in the collection (Atef, Mattar, Sherif, Elrefai, & Torki, 2020).Table 1 summarize the used datasets.

Table 1. Arabic Reading Comprehension Datasets

Reference	Name	Train	Test	Total size
(Mozannar, El Hajal, Maamary, & Hajj, 2019)	Arabic-SQuAD	38,885	9,459	48,344
(Mozannar, El Hajal, Maamary, & Hajj, 2019)	ARCD	695	700	1,395
(Clark et al., 2020)	TyDiQA- GoldP(Arabic)	14,724	921	15,645
(Atef, Mattar, Sherif, Elrefai, & Torki, 2020)	AQAD	4,108	1,151	5,259
	Merge all the datasets	58,493	11,270	69,763

5. Experiments and Evaluation

To provide a comprehensive analysis of pre-trained transformer-based models in Arabic QA, we evaluate three models, namely, AraBERT (AraBERTv0.2-large and AraBERTv2-base) and AraELECTRA on the four aforementioned datasets. Prior to the implementation, we first perform text pre-processing step to clean the textual data and remove word noise, and we perform text segmentation to meet the pre-trained model specifications.

5.1. Text Pre-processing

In preprocessing step, we applied the preprocessing methods adopted in a previous work of Antoun et al. (AUB Mind, n.d.) which performs the following: replace emojis, remove HTML markup except in TyDiQA-GoldP dataset, replace email URLs and mentions by special tokens remove diacritics and tatweel, insert whitespace before and after all non-Arabic digits, or English digits or Arabic and English Alphabet or the two brackets then, insert whitespace between words and numbers or numbers and words.

5.2. Text Segmentation

To follow the models' input specification, we used text pre-segmentation techniques for AraBERTv2-base, and no pre-segmentation techniques were required for AraBERTv0.2-large and AraELECTRA. AraBERTv2-base required the text to be pre-segmented by splitting the prefixes and suffixes from words, so we used Farasa segmenter (Abdelali, Darwish, Durrani, & Mubarak, 2016). The text tokenization was performed by applying the fast tokenization algorithm used by Farasa (Abdelali, Darwish, Durrani, & Mubarak, 2016). This tokenization has limitations (Antoun, Baly, & Hajj, 2020a), that result in providing different segmentation for similar answers and context. For example, when we pass some context and answer for the tokenization, it may segment them differently, resulting in different tokens for both contexts and answer so that the answer won't be recognized as a part of the context. The TyDiQA-GoldP dataset has 30 unrecognized pair of context/question, which we removed from the dataset. We also did extra preprocessing on the AQAD dataset, which was contained null values, so we removed these null values to ensure that will not affect the models' performance.

5.3. Dataset Splitting

To provide a valid comparison with other related studies, we used the original training and testing set of the ARCD,

AQAD, and TyDiQA-GoldP. For Arabic-SQuAD, we followed the author's splitting ratio into 10-10-80% (Mozannar, El Hajal, Maamary, & Hajj, 2019). We also followed previous work of Antoun et al. (Antoun, Baly, & Hajj, 2020a) and we trained on the whole Arabic-SQuAD with 50% of ARCD and test on the remaining 50% of ARCD.

5.4. Evaluation Metrics

We evaluated our different models based on two metrics which commonly used in QA tasks. The first is the exact match (EM), and the second is the F1-score.

5.4.1. F1 METRICS

F1 score is widely used in QA tasks, it is suitable when we care equally about precision and recall. It is calculated over the individual words in the prediction against those in the correct answer. Precision is the ratio of correctly predicted tokens divided by the number of all predicted tokens. The recall is also the ratio of correctly predicted tokens divided by the number of ground truth tokens. If a question has many answers, then the answer that provides the highest F1 score is considered as ground truth.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

5.4.2. Exact Match

This is a (true/false) metric that measures each question answer pair. If the predictions match the correct answers exactly, then EM = 1 else EM = 0. In a question, if the predicted answer and the correct answer are the same, then the score is 1, otherwise 0.

$$EM = \frac{\sum_{i=1}^{N} F\left(X_i\right)}{N} = \begin{cases} 1, \text{ if predicted answer = correct answer} \\ 0 = \text{otherwise} \end{cases}$$

5.5. Implementation Details

We implemented AraBERTv2-base, AraBERTv0.2-large, and AraELECTRA-base-discriminator on the reading comprehension datasets, namely Arabic-SQuAD, AQAD, TyDiQA-GoldP, ARCD, and the combined datasets.We searched for the best number of train epochs [2,3,4] and we tried different learning rate [1e-4, 2e-4, 3e-4, 5e-3]for fine-tuning, and we chose the hyper-parameters that gave us the best results based on the training set. We used the following hyper-parameters: four epochs and four batch size with a learning rate of 3e-5. The maximum total input sequence length after WordPiece tokenization is 384. The maximum number of tokens for the question is 64, and the maximum length of an answer that can be generated is 30. To provide a valid comparison, we used the same hyper-parameters on all the models.

6. Results and Discussion

We trained the AraBERTv2-base, AraBERTv0.2-large, and AraELECTRA on the Arabic-SQuAD training sets with the same hyper-parameters mentioned in section 5.5. The results from table 2 shows that our results were higher than mBERT. AraBERTv0.2-large achieved the best F1 and EM because we focused on training the model longer time. Running those experiments is computationally high, and the model takes around more than 12 h to train only for three epochs in our low-cost computation environment.The Arabic-SQuAD was affected by the mistranslation using Google Translate neural machine translation (NMT) (Mozannar, El Hajal, Maamary, & Hajj, 2019), and this explains the low results of this dataset.

Table 2. Comparison of the different text reader models on Arabic-SQuAD

Model	Arabic-SQuAD	
	F1	EM
mBERT (Mozannar, El Hajal, Maamary, & Hajj, 2019)	48.6	34.1
AraBERTv2-base (ours)	60.60	36.35
AraBERTv0.2-large (ours)	61.21	43.26
AraELECTRA (ours)	56.85	39.33

In the next experiments, we used ARCD to train our models. The results from table 3 shows big improvement in our models over mBERT.The best F1 and EM were achieved by AraELECTRA.The small size of ARCD was affect the performance of the models.

Table 3. Comparison of the different text reader models on ARCD

Model	ARCD	
	F1	EM
mBERT (Mozannar, El Hajal, Maamary, & Hajj, 2019)	50.10	23.9
AraBERTv2-base (ours)	59.25	26.21
AraBERTv0.2-large (ours)	56.08	23.64
AraELECTRA (ours)	68.15	35.47

Table 4 shows the results of training the AQAD dataset. Our models achieved a better F1-score compared to previous work (Atef, Mattar, Sherif, Elrefai, & Torki, 2020) but the EM value was less because the AQAD dataset is designed for questions that do not have answers.

Table 4. Comparison of the different text reader models on AQAD

Model	AQAD	
	F1	EM
mBERT (Atef, Mattar, Sherif, Elrefai, & Torki, 2020)	37	33
BIDAF Arabic FastText (Atef, Mattar, Sherif, Elrefai, & Torki, 2020)	32	32
AraBERTv2-base (ours)	40.32	19.32
AraBERTv0.2-large (ours)	40.23	25.54
AraELECTRA (ours)	39.18	24.85

On the experiments on the TyDiQA-GoldP dataset, we get the best result of F1 and EM compared to previous works that used the same models. In AraELECTRA, we noticed a decrease of F1 by 1%. For AraBERTv0.2-large on TyDiQA-GoldP, we record a 2% absolute increase in the exact match score over the previous work of Antoun et al. (Antoun, Baly, & Hajj, 2020b), which is the previous state-of-the-art. Exact Match (EM) measures the percentage of predictions that match any of the ground truth.

Table 5. Comparison of the different text reader models on TyDiQA-GoldP

Model	TyDiQA-GoldP(F1-EM)
mBERT (Clark et al., 2020)	81.7-
AraBERTv0.1 (AUB Mind, n.d.)	82.86-68.51
AraBERTv1 (AUB Mind, n.d.)	79.36-61.11
AraBERTv0.2-base (AUB Mind, n.d.)	85.41-73.07
AraBERTv2-base (AUB Mind, n.d.)	81.66-61.67
AraBERTv0.2-large (AUB Mind, n.d.)	86.03-73.72
AraBERTv2-large (AUB Mind, n.d.)	82.51-64.49
ArabicBERT-base (AUB Mind, n.d.)	81.24-67.42
ArabicBERT-large (AUB Mind, n.d.)	84.12-70.03
Arabic-ALBERT-base (AUB Mind, n.d.)	80.98-67.10
Arabic-ALBERT-large (AUB Mind, n.d.)	81.59-68.07
Arabic-ALBERT-xlarge (AUB Mind, n.d.)	84.59-71.12
AraELECTRA (AUB Mind, n.d.)	86.86-74.91
AraBERTv2-base (ours)	82.70-65.47
AraBERTv0.2-large (ours)	86.49-75.14
AraELECTRA (ours)	85.01-73.07

In our system, the AraBERTv0.2-large and AraELECTRA get the highest results reaching to 86 and 85 F1-score. In figure (Clark et al., 2020), we capture one of the results from the TyDiQA-GoldP development set. As you can see that the predicted answer match the exact ground truth answer.

Question	ما هو المسجد الحرام؟ What is the Holy Mosque?
Context	المسجد الحرام هو أعظم مسجد في الإسلام ويقع في قلب مدينة مكة غرب المملكة العربية السعودية، تتوسطه الكعبة المشرفة التي هي أول بيت وضع للناس على وجه الأرض ليعبدوا الله فيه تبعاً للعقيدة الإسلامية، وهذه هي أعظم وأقدس بقعة على وجه الأرض عند المسلمين. والمسجد الحرام هو قبلة المسلمين في صلاتهم، وإليه يحجون. سمي بالمسجد الحرام لحرمة القتال فيه منذ دخول النبي المصطفى إلى مكة المكرمة منتصراً. ويؤمن المسلمون أن الصلاة فيه تعادل مئة ألف صلاة. The Holy Mosque is the greatest mosque in Islam and is located in the heart of the city of Mecca, in the west of the Kingdom of Saudi Arabia, in the center of which is the Holy Kaaba, which is the first house placed for people on the earth to pray for Allah in it according to the Islamic belief, and this is the greatest and holiest spot on the earth for Muslims. It is the Qiblah of Muslims in their prayers, and to him they perform Hajj. It is called the Holy Mosque because of the sanctity of fighting there since the Prophet's Chosen One entered Mecca in victory. Muslims believe that a prayer in it is equivalent to a hundred thousand prayers.
Ground Truth	أعظم مسجد في الإسلام ويقع في قلب مدينة مكة غرب المملكة العربية السعودية. The greatest mosque in Islam and is located in the heart of the city of Mecca, in the west of the Kingdom of Saudi Arabia.
Predicted Answer	أعظم مسجد في الإسلام ويقع في قلب مدينة مكة غرب المملكة العربية السعودية. The greatest mosque in Islam and is located in the heart of the city of Mecca, in the west of the Kingdom of Saudi Arabia.

Fig. 3. Example of results from the TyDiQA-GoldP development set.

We also combined the ARCD and Arabic-SQuAD training set and fine-tuned the selected models. The results of AraBERTv2-base showed an improvement over the previous work of Antoun et al. (Antoun, Baly, & Hajj, 2020b) for about 3% F1 and about 2% EM and the other results almost the exact.

Table 6. Comparison of the text reader models on ARCD+Arabic-SQuAD

Model	ARCD+Arabic-SQuAD(F1-EM)
mBERT (Mozannar, El Hajal, Maamary, & Hajj, 2019)	61.3-34.2
AraBERTv0.1 (AUB Mind, n.d.)	67.45-31.62
AraBERTv1 (AUB Mind, n.d.)	67.8-31.7
AraBERTv0.2-base (AUB Mind, n.d.)	66.53-32.76
AraBERTv2-base (AUB Mind, n.d.)	67.23-31.34
AraBERTv0.2-large (AUB Mind, n.d.)	71.32-36.89
AraBERTv2-large (AUB Mind, n.d.)	68.12-34.19
ArabicBERT-base (AUB Mind, n.d.)	62.24-30.48
ArabicBERT-large (AUB Mind, n.d.)	67.27-33.33
Arabic-ALBERT-base (AUB Mind, n.d.)	61.33-30.91
Arabic-ALBERT-large (AUB Mind, n.d.)	65.41-34.19
Arabic-ALBERT-xlarge (AUB Mind, n.d.)	68.03-37.75
AraELECTRA (AUB Mind, n.d.)	71.22-37.03
AraBERTv2-base (ours)	70.38-33.33
AraBERTv0.2-large (ours)	70.75-35.19
AraELECTRA (ours)	71.51-37.18

In the last experiments to see the impact of fine-tuning on a larger dataset, we increased our training dataset by combining all four datasets in one JSON file. After fine-tuning in this training set for 3 epochs with the same hyperparameters, we get the results shown in Table 7. Compared to training the datasets separately, we found that on a large dataset, we get almost similar results and even lower results for some datasets. We noticed that as the models already gained knowledge of language-specific structure and semantics from pretraining, adapting them to the Arabic question answering task requires a small and useful dataset only to get overall great results.Table 8 shows a summary of our results from all experiments.

Table 7. Comparison of the different text reader models on the merged datasets

Model	Merge All The Datasets (F1-EM)
AraBERTv2-base (ours)	63.72-46.96
AraBERTv0.2-large (ours)	67.40-50.01
AraELECTRA (ours)	68.53-51.69

Table 8. Summary of our results on all datasets

Datasets	Models		
	AraBERTv2-base	AraELECTRA	AraBERTv0.2-large
Arabic-SQuAD	EM:36.35 F1: 60.60	EM: 39.33 F1:56.85	EM: 43.26 F1:61.21
ARCD	EM: 26.21 F1: 59.25	EM: 35.47 F1: 68.15	EM: 23.64 F1:56.08
AQAD	EM:19.32 F1:40.32	EM: 24.85 F1: 39.18	EM: 25.54 F1: 40.23
TyDiQA-GoldP(Arabic)	EM:65.47 F1:82.70	EM:73.07 F1: 85.01	EM:75.14 F1: 86.49
Arabic-SQuAD & ARCD	EM:33.33 F1:70.38	EM:35.19 F1:70.75	EM:37.18 F1:71.51
Merge all the datasets	EM: 46.96 F1:63.72	F1: 68.53 EM:50.01	EM :51.69 F1:67.40

Table 9. Comparison of the different text reader models on different datasets

	Datasets											
	Arabic-SQuAD		ARCD		ARCD&Arabic-SQuAD		TyDiQA-GoldP		AQAD		Merged Datasets	
Models	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
mBERT (Mozannar, El Hajal, Maamary, & Hajj, 2019) (Clark et al., 2020) (Atef, Mattar, Sherif, Elrefai, & Torki, 2020)	48.6	34.1	50.1	23.9	61.3	34.2	81.7	-	37	33	-	-
BIDAF Arabic FastText (Atef, Mattar, Sherif, Elrefai, & Torki, 2020)	-	-	-	-	-	-	-	-	32	32	-	-
AraBERTv0.1 (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	67.45	31.62	82.86	68.51	-	-	-	-
AraBERTv1 (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	67.8	31.7	79.36	61.11	-	-	-	-
AraBERTv0.2-base (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	66.53	32.76	85.41	73.07	-	-	-	-
AraBERTv2-base (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	67.23	31.34	81.66	61.67	-	-	-	-
AraBERTv2-base (ours)	60.60	36.35	59.25	26.21	70.38	33.33	82.70	65.47	40.32	19.32	63.72	46.96
AraBERTv0.2-large (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	71.32	36.89	86.03	73.72	-	-	-	-
AraBERTv0.2-large (ours)	61.21	43.26	56.08	23.64	71.51	37.18	86.49	75.14	40.23	25.54	67.40	51.69
AraBERTv2-large (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	68.12	34.19	82.51	64.49	-	-	-	-
ArabicBERT-base (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	62.24	30.48	81.24	67.42	-	-	-	-
ArabicBERT-large (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	67.27	33.33	84.12	70.03	-	-	-	-
Arabic-ALBERT-base (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	61.33	30.91	80.98	67.10	-	-	-	-
Arabic-ALBERT-large (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	65.41	34.19	81.59	68.07	-	-	-	-

	Datasets											
	Arabic-SQuAD		ARCD		ARCD&Arabic-SQuAD		TyDiQA-GoldP		AQAD		Merged Datasets	
Models	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
Arabic-ALBERT-xlarge (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	68.03	37.75	84.59	71.12	-	-	-	-
AraELECTRA (Antoun, Baly, & Hajj, 2020b)	-	-	-	-	71.22	37.03	86.86	74.91	-	-	-	-
AraELECTRA(ours)	56.85	39.33	68.15	35.47	70.75	35.19	85.01	73.07	39.18	24.85	68.53	50.01

Table 9 summarizes all previous works results compared to our results; using the Arabic-SQuAD dataset, we were able to achieve the best results using AraBERTv0.2-large.For the ARCD dataset, we achieved the best results using our models compared to mBERT.Using both ARCD and Arabic-SQuAD training set, our AraBERTv0.2-large model achieved a small increase in F1, but the Arabic-ALBERT-xlarge (Antoun, Baly, & Hajj, 2020b) did achieve a higher tiny increase in EM than our models. The low results of ARCD and Arabic-SQuAD are due to the poor quality of the training examples, which are translated from English SQuAD. ARCD training set also contained text in languages other than Arabic, which can reduced performance due to the unknowns subwords and characters (Antoun, Baly, & Hajj, 2020b).

On the experiments on TyDiQA-GoldP dataset using our AraBERTv0.2-large, we got the best result of EM compared to previous work models. Although in F1, we did not achieve that increase. The results on this dataset were much higher than others datasets. We believe that because the dataset is much more cleaner and correctly labeled without any translation. This dataset is created by experts in the Arabic language. We recognize that the deep understanding of the data itself is the key to understanding what modeling techniques will be better suited. Our experiments on the AQAD dataset obtained higher results than previous models. However, compared to other datasets, the AQAD dataset always got low results. We believe the quality of the dataset can affect the performance of the models.

6.1 Explaining Models Performance

Question Answering (QA)task used to identifying the correct span in the passage that answers the question. QA systems may have incorrect results, and it is important to the end-user to know the reasons. To address this, we use a gradient-based explanation approach to show the model behavior and which parts of a sentence are used to predict the answer. The gradient-based explanation approach is used by leverage the gradients in a trained deep neural network to explain the relation between inputs and output. The gradient identifies how much a change in every input would change the predictions (TF a2.0 Tutorial, n.d.). We implemented the following steps to explain AraBERTv0.2-large behavior trained on TyDiQA-GoldP and AQAD datasets: Create one representation of each input. Multiply one representation by model embedding matrix then feed into the model to get a prediction for start and end span positions. Get gradient of correct start and end span position.

From the figures 4 and 5, we noticed that AraBERTv0.2-large trained on TyDiQA-GoldP returns the correct answers, but on the AQAD dataset, it does not.

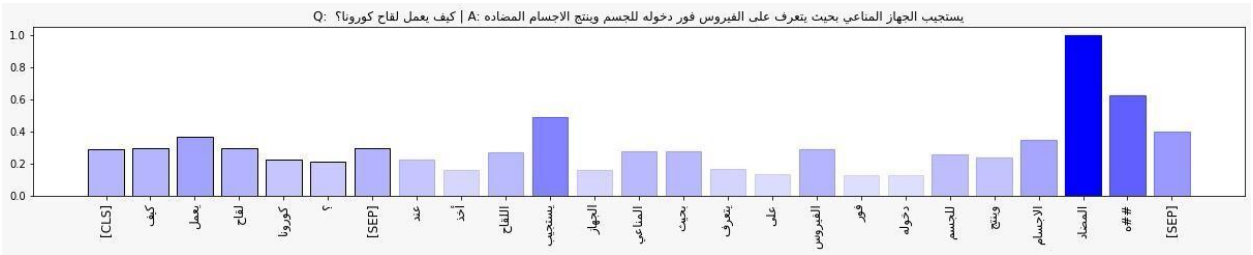

Fig. 4. Example of results from AraBERTv0.2-large trained on TyDiQA-GoldP.

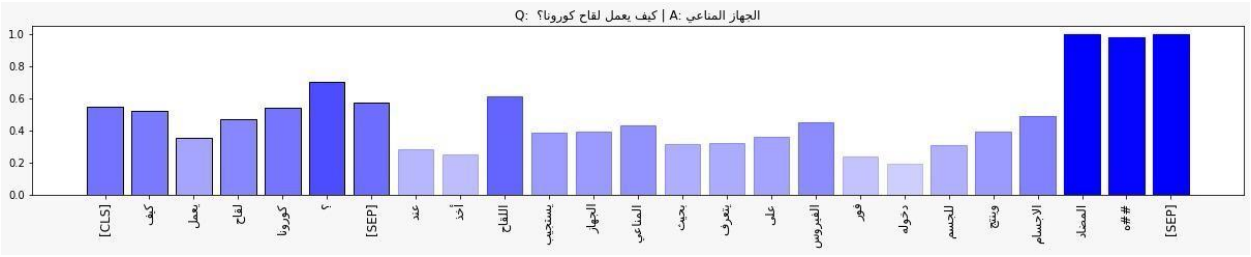

Fig. 5. Example of results from AraBERTv0.2-large trained on AQAD.

7. Conclusion

Question Answering is an important research area in the NLP field; the goal of a QA system is to answers the questions written in natural language. The current growth of language models like BERT and ELECTRA has made it possible for all kinds of NLP tasks to make significant progress. In this paper, we evaluate the performance of three existing Arabic pre-trained models on Arabic QA. For our QA system, a model is trained to answer questions from the given passage. The AraBERT and AraELECTRA trained in the context of Question Answering with Arabic-SQuAD, ARCD, AQAD, and TyDiQA-GoldP datasets. Our experiments address a comprehensive study of different QA models for Arabic and how the results will be affected by different factors.

References

Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016). Farasa: A fast and furious segmenter for Arabic. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 11–16. https://doi.org/10.18653/v1/N16-3003

Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools with a Shared Task on Offensive Language Detection*, 9–15. https://doi.org/10.48550/arXiv.2003.00104

Antoun, W., Baly, F., & Hajj, H. (2020). AraELECTRA: Pre-training text discriminators for Arabic language understanding. *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, 191–195. https://doi.org/10.18653/v1/2020.wanlp-1.21

Atef, A., Mattar, B., Sherif, S., Elrefai, E., & Torki, M. (2020, November). AQAD: 17,000+ Arabic questions for machine comprehension of text. *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, 1–7. https://doi.org/10.1109/AICCSA50499.2020.9316453

Biltawi, M. M., Tedmori, S., & Awajan, A. (2021). Arabic question answering systems: Gap analysis. *IEEE Access, 9*, 63876–63904. https://doi.org/10.1109/ACCESS.2021.3074692

Church, K. W. (2017). Emerging trends: Word2Vec. *Natural Language Engineering, 23*(1), 155–162. https://doi.org/10.1017/S1351324916000335

Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Palomaki, J., … & Xia, P. (2020). TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics, 8*, 454–470. https://doi.org/10.1162/tacl_a_00317

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://doi.org/10.48550/arXiv.1810.04805

Ganin, I., Goodfellow, I., & others. (2020). Generative adversarial networks. *Communications of the ACM, 63*(11), 139–144. https://doi.org/10.1145/3422622

Hiemstra, D. (2000). A probabilistic justification for using tf–idf term weighting in information retrieval. *International Journal on Digital Libraries, 3*(2), 131–139. https://doi.org/10.1007/s007999900018

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1909.11942

Liao, M., Shi, B., Bai, X., Wang, X., & Yao, C. (2017). TextBoxes: A fast text detector with a single deep neural network. *Proceedings of the AAAI Conference on Artificial Intelligence, 31*(1). https://doi.org/10.1609/aaai.v31i1.10781

McCormick, C. (2019). Question answering with a fine-tuned BERT. *Chris McCormick Blog*. https://mccormickml.com/2019/07/22/question-answering-bert/

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in pre-training distributed word representations. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. https://doi.org/10.48550/arXiv.1712.09405

Mishra, A., & Vishwakarma, S. (2015). Analysis of TF-IDF model and its variant for document retrieval. *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, 749–753. https://doi.org/10.1109/CICN.2015.158

Mozannar, H., El Hajal, K., Maamary, E., & Hajj, H. (2019). Neural Arabic question answering. *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, 108–118. https://doi.org/10.18653/v1/W19-4612

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. https://doi.org/10.18653/v1/D16-1264

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*. https://doi.org/10.48550/arXiv.1706.03762

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., … & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint*. https://doi.org/10.48550/arXiv.1609.08144

Zhang, Y., Jin, R., & Zhou, Z.-H. (2010). Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics, 1*(1-4), 43–52. https://doi.org/10.1007/s13042-010-0001-0