IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 May 2022, accepted 23 June 2022, date of publication 12 July 2022, date of current version 21 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3189956

RESEARCH ARTICLE

An ELECTRA-Based Model for Neural Coreference Resolution

FRANCESCO GARGIULO[1], ANIELLO MINUTOLO[1], RAFFAELE GUARASCI[1], EMANUELE DAMIANO[1], GIUSEPPE DE PIETRO[1], HAMIDO FUJITA[2,3,4], (Senior Member, IEEE), AND MASSIMO ESPOSITO[1]

[1]Institute for High Performance Computing and Networking (ICAR), National Research Council (CNR), 80131 Naples, Italy
[2]Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh City 70000, Vietnam
[3]Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, 18011 Granada, Spain
[4]Faculty of Software and Information Science, Iwate Prefectural University, Takizawa, Iwate 020-0693, Japan

Corresponding author: Raffaele Guarasci (raffaele.guarasci@cnr.it)

ABSTRACT In last years, coreference resolution has received a sensibly performance boost exploiting different pre-trained Neural Language Models, from BERT to SpanBERT until Longformer. This work is aimed at assessing, for the first time, the impact of ELECTRA model on this task, moved by the experimental evidence of an improved contextual representation and better performance on different downstream tasks. In particular, ELECTRA has been employed as representation layer in an assessed neural coreference architecture able to determine entity mentions among spans of text and to best cluster them. The architecture itself has been optimized: i) by simplifying the modality of representation of spans of text but still considering both the context they appear and their entire content, ii) by maximizing both the number and length of input textual segments to exploit better the improved contextual representation power of ELECTRA, iii) by maximizing the number of spans of text to be processed, since potentially representing mentions, preserving computational efficiency. Experimental results on the OntoNotes dataset have shown the effectiveness of this solution from both a quantitative and qualitative perspective, and also with respect to other state-of-the-art models, thanks to a more proficient token and span representation. The results also hint at the possible use of this solution also for low-resource languages, simply requiring a pre-trained version of ELECTRA instead of language-specific models trained to handle either spans of text or long documents.

INDEX TERMS Coreference resolution, ELECTRA, neural language model, OntoNotes, natural language processing.

I. INTRODUCTION

Coreference resolution is the task aimed at determining referring expressions (mentions) - are either pronouns, noun phrases or named entities - that point to the same real-world entities (referents) in a document. Although it is one of the oldest tasks in Natural Language Processing (NLP), it still cannot be considered as solved, experiencing several different approaches over years, from early rule-based and statistical to machine and deep learning ones [1]–[3]. More recently, Neural Language Models (NLMs) have been proposed, obtaining sensibly improvements on different semantic benchmarks, such as question-answering, natural language inference, and named entity recognition [4]. These NLMs are pre-trained on large text corpora to obtain a general-purpose language representation and, then, fine-tuned for a down-stream task, by only re-training a single, task-specific layer at the output.

This approach has been widely used in almost all the NLP tasks, except for the coreference resolution, where, in the last decade, sophisticated end-to-end architectures have been proposed, where NLMs like ELMo (*Embeddings from Language Models*) [5], [6] and BERT (*Bidirectional Encoder Representations from Transformers*) [7]–[9] have been utilized to numerically encode the tokens composing the input sentences, reaching significant performance gains over all previous more traditional approaches [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

The most used and re-adapted end-to-end coreference architecture is represented by the one proposed in [5], [11], that jointly learns which spans of text in a document are entity mentions and how to best cluster them. This architecture reasons over the space of all spans, taking into account computational limits, and directly optimizes, for each span, the marginal likelihood of having a good antecedent in the previous ones [11]. A core element of this architecture is the representation layer aimed at first encoding input tokens with numerical embeddings and, thus, obtaining the consequent span embeddings by combining them.

To improve the representation abilities of this architecture with respect to the original proposal, different contextualized NLMs, from BERT [4] to SpanBERT [9] until Longformer [12], have been adopted, reaching a continuous gain in the final performance [8], [9], [13].

This work is framed in this context, and, in particular, is aimed at adopting, for the first time, the novel ELECTRA model (*Efficiently Learning an Encoder that Classifies Token Replacements Accurately*) [14] as representation layer in the above-mentioned coreference architecture. Reasons for this choice lie in better performance shown by ELECTRA in capturing contextual word representations, also outperforming other NLMs, like BERT, in downstream tasks [15].

Moreover, the architecture itself has been optimized: i) by simplifying the modality of representation of spans of text, but still considering both the context they appear and their entire content, ii) by maximizing both number and length of input textual segments to better exploit the representation power of ELECTRA as contextualized NLM, iii) by maximizing the number of spans of text to be processed, since potentially representing mentions, following computational limits as well as training dataset statistics.

This ELECTRA-powered model has been trained and evaluated on the OntoNotes dataset [16] from both a quantitative and qualitative perspective, comparing its performance with respect to the state-of-the-art and deeply analyzing its behavior in terms of typology and length of mentions that are correctly predicted.

The paper is organized as follows. Section 2 reviews the state-of-the art of approaches, also giving details on existing neural language models. Section 3 outlines the research objective and contribution, also in comparison with existing state-of-the-art approaches. Sections 4 details the proposed solution, also given the description of both the dataset and the neural language model used. Section 5 outlines the experimental assessment, whereas results are presented, analyzed and discussed in the Section 6, from both a quantitative and qualitative perspective. Section 7 concludes the work.

II. LITERATURE REVIEW

In this section the scientific articles that form the foundation of this paper are detailed. In particular, an overview of coreference resolution approaches and systems is provided in section II-A, whereas in section II-B, an outline of neural language modelling methods is given.

A. COREFERENCE APPROACHES

Coreference resolution has been one of the historical NLP tasks, therefore many approaches have been proposed over the years, since rule-based to statistical ones until to deep learning-based [1].

The first neural coreference resolution model [17] has been focused on two critical aspects: the identification of non-anaphoric references in texts and the ability to distinguish mentions from non-mentions. Later developed models [18], [19] have incorporated entity-level information produced by a Recurrent Neural Network (RNN) in order to exploit global features about entity clusters. These models have relied on including features defined on mention-pair clusters. Another approach [20] has exploited a neural mention ranking model [19] in order to replace the heuristic loss functions with reinforced-learning based policy gradient algorithm.

Currently, the state-of-the-art approach is represented by the *end-to-end coreference* (*e2e-coref*) model [11]. It has been based on the construction of high-dimensional word embeddings to represent tokens of annotated documents. Spans of text in the document are represented by combining context-dependent boundary representations with a head-finding attention mechanism over all their tokens. Although difficult to maintain because of its high-dimensionality, this system, based on Long Short-Term Memory (LSTM) networks, has, as its strengths, the ability to capture long term dependencies. An evolution of this work has been the *coarse-to-fine coreference* (*c2f-coref*) model [5], using ELMO for word representation, but without altering the modality of representation of spans of text.

More recent approaches have re-adapted *c2f-coref* model by first substituting ELMO with BERT to enhance the representation abilities [8]. In particular, the entire LSTM-based encoder has been replaced with BERT, representing a span of text in terms of the first and last word-pieces, concatenated with the attended version of all word pieces occurring in it. Then, SpanBERT [9] has been exploited in place of BERT, since specifically re-trained to better represent and predict spans of text, thus most fitting with the need of representing multi-token mentions. Other modifications have been proposed to advance *c2f-coref* model, like attended antecedent, entity equalization, span clustering, and cluster merging. Still, they have been proved to not improve performance [7], [10], with a negative to marginal impact.

Similar performance to the *c2f-coref* model enhanced with SpanBERT but with lower memory usage has been achieved by the *start-to-end coreference* (*s2e-coref*) model [13]. This approach has been aimed at not constructing span representations but at utilizing the endpoints of a span (rather than all span tokens) to compute mention and antecedent scores through a series of bilinear functions over their contextualized representations. Moreover, SpanBERT has been substituted with Longformer [12], in order to exploit its better ability to process long documents without resorting to sliding windows or truncation.

Among the many recent approaches derived from *e2e-coref* model, it is noteworthy the one proposed by [21], named *Simplified e2e-coref* model, offering a simplified version of the original neural model. Most lately, a new model, named *CorefQA*, has been proposed [22], based on SpanBERT and handling the task of coreference resolution as an extractive question-answering one, achieving a further boost in the performance but at the cost of a worsening of the execution time.

B. NEURAL LANGUAGE MODELS

The most used NLM in coreference resolution task is BERT [4], which is also the most widely used model in the whole NLP field. BERT is usually pretrained on a very large unannotated text corpus, exploiting Masked Language Modelling (MLM) approach, which consists in randomly applying a mask on a percentage of words in the training corpus. This allows encoding information from both directions of the sentences and training the model to predict the masked words. The input vocabulary can be *cased* or *uncased*, leading to two different pretrained models. Because of the time and computational resources demanded for training, in recent years several pretrained BERT models have been made available in literature.

A NLM extending BERT capabilities and developed for tasks such as coreference resolution and question answering is SpanBERT [9]. It is a method for span-based pre-training that outperforms BERT on two ways: masking contiguous random spans, not only random tokens, and training the span boundary representations to predict the entire content of the masked span, without relying on the individual token representations.

Longformer [12] is a modified Transformer architecture with a self-attention operation able to scale linearly instead of quadratically with the sequence length, making it able to process long documents. This allows overcoming the limits of existing approaches to handle text segments made of, at most, 512 tokens characterizing BERT-style pretrained NLMs. Thus, using multiple layers of attention, it is possible to build contextual representations of the whole context limiting the need for task-specific architectures. Longformer has been successfully applied to different downstream NLP tasks requiring document-level processing abilities.

A new NLM worthy of attention is ELECTRA [14], which has demonstrated to reach or even exceed performance of BERT-based models using less compute resources. It uses two Transformer models, a generator and a discriminator. Unlike other NLMs, the pre-training task in ELECTRA is based on Replaced Token Detection (RTD) approach that corrupts the input replacing some tokens with plausible alternatives sampled from generator network. This allows the model to learn from all input tokens and not just from a masked subset. As far as is known, ELECTRA has never yet been used for coreference resolution task.

III. RESEARCH OBJECTIVE AND CONTRIBUTION

The main objective of this work is to investigate the use of ELECTRA to improve performance for the task of coreference resolution, moved by the experimental evidence of an improved contextual representation able to substantially outperform the ones learned by other NLMs, achieving higher accuracy on different downstream tasks.

To this aim, first of all, ELECTRA has been used in *c2f-coref* architecture, in place of BERT and SpanBERT, in order to encode input tokens and improve their contextualized representation.

Secondly, spans of text have been chosen to be represented by continuing to use all their tokens, differently from the recent proposal of [13] where span representations are not built, and only the endpoints of a span (rather than all span tokens) are used to compute mention and antecedent scores. In such a way, spans of text are still represented by considering both the context they appear in, i.e. exploiting contextualized embeddings of their boundary tokens, and their entire content, averaging embeddings of all their tokens. This choice allows avoiding codifying spans of text only in terms of their boundary tokens since these latter could be uninformative to approximate the whole span content, for instance, in case they are articles, prepositions or adjectives. Moreover, differently from the original architecture [11] and the other successive evolutions [5], [8], [9], a simpler and more computationally efficient way has been adopted to exploit the contribution of all the tokens composing a span, i.e. using the average instead of a head-finding attention mechanism, where all its weights have to be learned during the training process.

Furthermore, both the number and the length of textual segments given in input to *c2f-coref* architecture have been optimized in accordance with limits of the transformer architecture [23] at the basis of ELECTRA and in order to preserve the computational efficiency, maximize the coverage of the input documents, thus producing a reduced truncation for them, and exploit in the best possible way the representation power of ELECTRA as contextualized NLM. Even though recent work on Longformers has allowed shifting towards processing complete documents, they admit only a smaller model, i.e. in its base version, and with one training example per batch [13].

Finally, the number of spans of text to be processed as input of *c2f-coref* architecture, since potentially representing mentions, has been also maximized by considering input dataset statistics, such as the maximum mention width or mentions-tokens ratio for documents or partitions of them, but also not violating computational limits due the high spatial complexity of *c2f-coref* architecture, which is equal to $O(n^4)$. This choice is motivated by the consideration that the larger the number of spans evaluated, the more likely it is to identify potential mentions in them.

All these aspects represent the main contributions of this work, that will be deeply investigated in the following sections.

IV. DATASET AND METHODS

The working process behind the proposed coreference resolution system can be summarized as follows.

First, documents composing the OntoNotes dataset are given as input. Secondly, the end-to-end *c2f-coref* architecture proposed by [5], [11] and powered by the usage of ELECTRA to represent input tokens is leveraged to calculate the coreference predictions. A detailed description of the OntoNotes dataset (Section IV-A), of the system architecture (Section IV-B) and the ELECTRA model (Section IV-C) is provided below.

A. DATASET

OntoNotes is the most used corpus in coreference resolution tasks [2]. It is part of the OntoNotes project, aiming at the creation of a multilingual corpus with multiple level of annotation. It includes three languages (English, Chinese and Arabic), collecting texts from several domains manually annotated with syntactic structure and shallow semantics [16]. For the purpose of this work, only the English sub-corpus has been taken into account.

OntoNotes is divided into three subsets (*Train*, *Dev*, and *Test*), which, respectively, can be used for training, developing and testing a coreference model. The subsets are arranged into sets of documents, each of which is composed of an ordered list of non-overlapping partitions of ordered utterances. Statistics on the dataset are reported in Table 1.

TABLE 1. OntoNotes statistics.

Measure	Train	Dev	Test
Partitions for document	1.44	1.55	1.57
Tokens for partition	463.7	475.52	487.3
Mentions for partition	55.52	55.85	56.79
Coreference clusters for partition	12.54	13.25	13.024
Mentions for cluster	4.43	4.21	4.36
Mention width	2.28	2.37	2.28
Total mentions count	155560	19156	19764
Total coreference clusters count	35143	4546	4532
Total partitions count	2802	343	348
Total documents count	1940	222	222

OntoNotes presents different layers of annotation, including syntax, propositions, word sense, named entities and coreference. The latter is the one used in this work and considers every type of potential mentions: pronouns (PRP), noun phrases (NP) and verb phrases (VP).

With respect to the width of mentions, most mentions are composed of one (in detail, 59.92% for *Train*, 58.08% for *Dev* and 59.74% for *Test*) or two tokens (respectively, 79.11%, 78.02% and 79.45%), but there are many very long ones, reaching up to 94 tokens (respectively, 94, 63 and 58).

Going into detail, Figure 1 shows the mentions distribution in terms of cumulative histogram over mentions width. In particular, the cumulative histogram highlights what percentage of the total number of mentions has a width no greater than that considered. It is worth noting that almost the totality of all the mentions (respectively, about 99.96%, 99.96% and 99.97%) presents a width of at most 40 tokens.

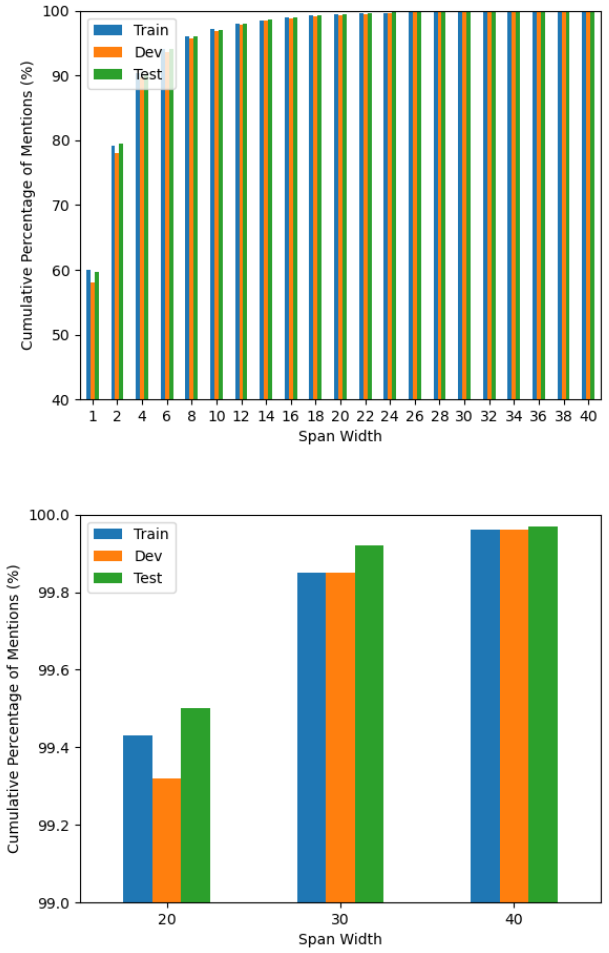


FIGURE 1. The histograms represent the cumulative distribution of mentions over their span width for the Train, Dev and Test sets. The figure on the top shows all the values from 1 to 40, on the other hand, the bottom histogram shows only a zoom for the values (20, 30, 40).

On average, partitions contain more than 400 tokens (in detail, 463.7 for *Train*, 475.52 for *Dev* and 487.3 for *Test*), while the number of mentions contained in a partition is about 55.52, 55.85 and 56.79, respectively.

It is worth noting that the ratio between i) the number of token composing a partition and ii) the number of spans (i.e. ordered sequences of tokens) which can be successfully recognized as mentions within that partition, is about 0.12, on average. Going into detail, as reported in Figure 2, all partitions present a ratio less or equal of 0.3. As a consequence, the maximum number of antecedents a mention can have is at most about 0.3 * number of tokens within the same partition.

B. SYSTEM ARCHITECTURE

The neural architecture here used is the *c2f-coref* model proposed by [5], [11]. In accordance with this architecture, the task of coreference resolution is defined as a set of antecedent assignments y_i for each span i, with $1 \leq i \leq N$, belonging to a given document D that contains T tokens and $N = \frac{T(T+1)}{2}$ possible text spans.

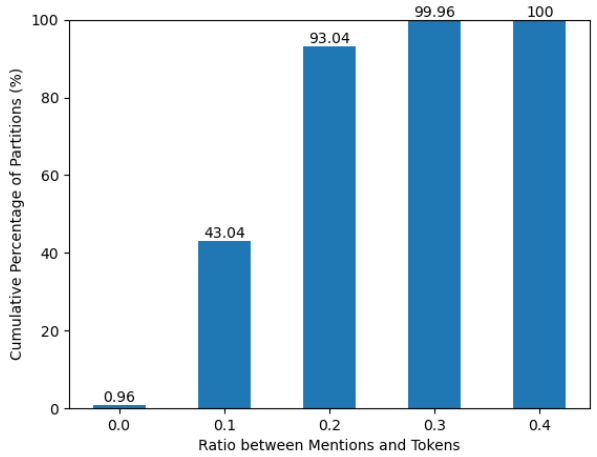


FIGURE 2. The distribution of partitions for mentions-tokens ratio.

In particular, all spans are considered as potential mentions and, for each span i, the set of antecedent assignments y_i, i.e. mentions preceding the span under examination and referring to the same entity, is calculated. The set of of possible assignments for each y_i is $Y(i) = \{\epsilon, 1, \ldots, i-1\}$, which includes the dummy antecedent ϵ and all preceding spans.

The dummy antecedent covers the cases when a span is not an entity mention, or it is an entity mention but is not coreferent with other previous spans.

Grouping all spans connected by a set of antecedent predictions allows defining a final clustering.

To realize this task, *c2f-coref* model learns a conditional probability distribution $P(y_1, \ldots, y_n|D)$ whose most likely configuration corresponds to the correct clustering. This distribution is calculated as the product of multinomials for each span:

$$P(y_1, \ldots, y_N|D) = \prod_{i=1}^{N} P(y_1|D) = \prod_{i=1}^{N} \frac{exp(s(i, y_j))}{\sum_{y' \in Y(i)} exp(s(i, y'))} \quad (1)$$

where $s(i, j)$ is a pairwise score for a coreference link between span i and span j in document D. This coreference score is computed as follows:

$$s(i, j) = \begin{cases} 0 & j = \varepsilon \\ s_m(i) + s_m(j) + s_a(i, j) & j \neq \varepsilon \end{cases} \quad (2)$$

It is equal to 0 in case of dummy antecedent, otherwise it is the sum of three terms, namely $s_m(i)$ and $s_m(j)$ are the scores indicating that the spans i and j are mentions, and $s_a(i, j)$ is the score indicating that the span j is an antecedent for the span i.

The model predicts the best antecedent score if all non-dummy scores are positive, otherwise it vanishes. Differently from the original *c2f-coref* model, each span i is given an embedding representation h_i by using ELECTRA as shown in Figure 3 and described in the next subsection.

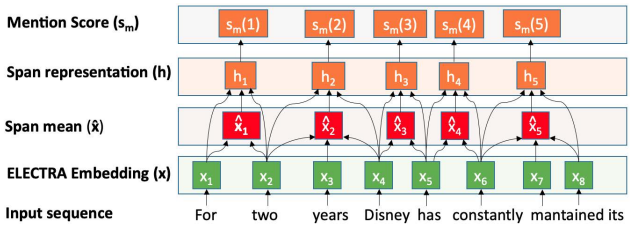


FIGURE 3. Span representation using ELECTRA and mean of the internal span words.

Given these span representations, the scoring functions s_m and s_a are calculated, as shown in Figure 4, via feed-forward neural networks $FFNN_m$ and $FFNN_a$ as follows:

$$s_m(i) = w_m \cdot FFNN_m(h_i) \quad (3)$$

$$s_a(i, j) = w_a \cdot FFNN_a([h_i, h_j, h_i \circ h_j, \phi(i, j)]) \quad (4)$$

where $\cdot$ is the dot product; $\circ$ is element-wise multiplication; $FFNN$ is a feed forward neural network calculating a non-linear mapping from input to output; $s_a(i, j)$ includes explicit element-wise similarity of each span e_i and a feature vector $\phi(i, j)$ containing information about speaker, genre and other syntactic metadata.

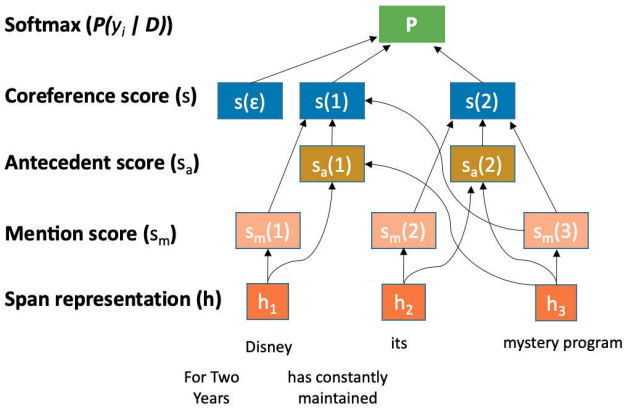


FIGURE 4. Calculation of mention and antecedent scoring functions.

For the training, the marginal log-likelihood of all correct antecedents implied by the gold clustering is optimized:

$$log \prod_{i=1}^{N} \sum_{\hat{y} \in Y(i) \cap GOLD(i)} P(\hat{y}) \quad (5)$$

where $GOLD(i)$ is the set in the gold cluster containing the span i.

A challenging characteristic of this model is its space complexity of $O(T^4)$ in the number of input tokens T. In order to grant computational efficiency, a pruning strategy has been adopted to reduce the number of spans representing candidate mentions to be processed.

Specifically, first, candidate mentions are pruned depending on their width expressed by the *max_span_width* hyperparameter.

Then, their number is further reduced by considering only M spans with the highest mention scores $s_m(\cdot)$, where M is an heuristic value that is dynamically calculated as product of the number of tokens T of the document and a cutting percentage *top_span_ratio*. Finally, for the remaining spans, only up to K antecedents are considered for each one, where K is determined by fixing the *max_top_antecedents* hyperparameter.

In order to determine the top antecedents to consider for each span, the *coarse-to-fine antecedent pruning* strategy proposed by [5] has been used, aimed at estimating the aforementioned $s_a(\cdot,\cdot)$ using an alternate bi-linear scoring function that can be learned in an end-to-end fashion. By using this approach, *max_top_antecedents* can be fixed equal to 50, with comparable performance that can be obtained without it, but with a less aggressive pruning, so considering up to 250 antecedents per span, and thus worsening the memory occupation and computational efficiency.

C. ELECTRA

ELECTRA NLM relies on two Transformer models, that share the same word embedding, namely a generator G and a discriminator D, and it is based on training D to distinguish *fake* or replaced input tokens produced by G in the sequence. This approach, called replaced token detection (RTD), allows using a minor number of examples without losing in performance.

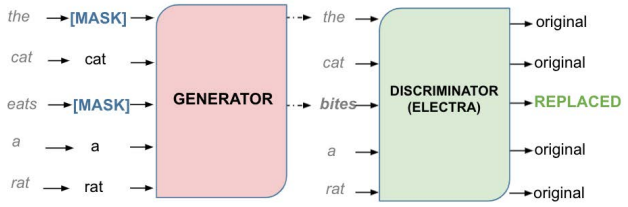


FIGURE 5. ELECTRA overview with replaced token detection. The generator *G* is usually a MLM trained with the discriminator *D* but it may virtually be any model producing an output distribution over tokens.

In particular, for a given input sequence, where some tokens are randomly replaced with a [MASK] token, G is trained to predict the original tokens for all masked ones. On the other hand, G is given input sequences built by replacing [MASK] tokens with *fake* ones produced by G, and it is trained to predict whether they are original or *fake*.

More formally, given an input sentence s of raw text χ, composed by a sequence of tokens $s = w_1, w_2, \ldots, w_n$ where w_t $(1 \leq t \leq n)$ represents the generic token, both G and D firstly encode s into a sequence of contextualized vector representations $h(s) = h_1, h_2, \ldots, h_n$.

Then, for a given position t so that the corresponding $w_t = [MASK]$, the generator outputs the probability to have a token w_t, with a softmax layer:

$$p_G(w_t|s) = \frac{e(w_t)^T h_G(s)_t}{\sum_{w'} exp(e(w')^T h_G(s)_t)} \tag{6}$$

where $e(\cdot)$ represents the embedding function.

On the other hand, the discriminator predicts whether w_t is the original or "fake", using a sigmoid layer:

$$D(s,t) = sigmoid(e(w_t)^T h_D(s)_t) \tag{7}$$

During the pre-training, G employs the following loss function:

$$\mathcal{L}_{Gen} = \mathcal{L}_{MLM} = \mathbb{E}(\sum_{i \in m} -\log p_G(w_i|s^{masked})) \tag{8}$$

where $m = m_1, m_2, \ldots, m_k$ are k random selected words and s^{masked} is the sentence with the masked words.

On the other hand, D uses the following loss function:

$$\mathcal{L}_{Dis} = \mathbb{E}(\sum_{t=1}^{n} -\mathbb{I}(w_t^{corrupt} = x_t) \log D(s^{corrupt}, t) + -\mathbb{I}(w_t^{corrupt} \neq x_t) \log D(s^{corrupt}, t)) \tag{9}$$

where $w_t^{corrupt}$ is the corrupted word within the corrupted sentence $s^{corrupt}$.

Finally, the following combined loss is minimized:

$$\min_{\theta_G, \theta_D} \sum_{s \in \chi} \mathcal{L}_{Gen}(s, \theta_G) + \lambda \mathcal{L}_{Dis}(s, \theta_D) \tag{10}$$

At the end of the pre-training, G is discarded and only D is used.

The main reason for which ELECTRA efficiency results improved with respect to BERT-like NLMs is that predictions are calculated not only over masked tokens, but also for each token and the discriminator loss can be calculated over all input tokens.

In the *c2f-coref* model here adopted, ELECTRA is used to represent each span by considering both the context they appear, i.e. exploiting contextualized embeddings of its boundary tokens, and its entire content, averaging embeddings over all its tokens. In more detail, for each span i, its representation h_i obtained by using ELECTRA is given by:

$$h_i = [x^*_{START(i)}, x^*_{END(i)}, \hat{x}_{SPAN(i)}, \phi(i)\,] \tag{11}$$

where $x^*_{START(i)}$ and $x^*_{END(i)}$ are the embedding representations of the boundary tokens, $\hat{x}_{SPAN(i)}$ is the embedding representation of the whole span, calculated by averaging the embedding representations of all its tokens, and $\phi(i)$ is a vector feature which encodes the span size.

More formally, $\hat{x}_{SPAN(i)}$ is defined as follows:

$$\hat{x}_{SPAN(i)} = \frac{1}{dim(SPAN(i))} \sum_{t=START(i)}^{END(i)} \cdot x_t \tag{12}$$

where $\hat{x}_{SPAN(i)}$ is the mean of the embedding representations x_t belonging to the span i and $dim(SPAN(i))$ is the number of span tokens.

Moreover, following the same strategy proposed in [8], each input document has been split into non-overlapping segments of length determined by the *max_segment_length* hyperparameter. For ELECTRA, BERT and SpanBERT this hyperparameter has an upper-bound of 512, determined by

the usage of transformer architecture. Thus, the embedding representation for each token is calculated depending on the set of words that lie in the same segment.

Finally, following [5], a batch size of one document has been used, with a number of training segments, determined by the *max_training_ sentences* hyperparameter, chosen hand in hand with the *max_segment_ length* hyperparameter in order to preserve the computational efficiency, due to the memory intensiveness of span representations, maximize the coverage of the input document, with a reduced truncation, and, exploit, in the best possible way, the representation power of ELECTRA as contextualized NLM.

V. EXPERIMENTAL SETUP AND METRICS

Hereafter the experimental setup and the evaluation metrics are described in Section V-A and V-B respectively.

A. EXPERIMENTAL SETUP

The experimental assessment of the *c2f-coref* model powered by ELECTRA, as described in the previous section, has been arranged by exploiting the implementation[1] released by [10], by integrating ELECTRA to calculate span representations. ELECTRA model in its large (cased) version[2] has been used, which is made available by Hugging Face Transformers[3] framework. This framework provides state-of-the-art Transformer-based architectures with thousands of pre-trained models in over a hundred languages for NLP tasks. In particular, this specific ELECTRA model has been pretrained on a dataset which is 33B tokens greater than the one used for BERT, by including data from ClueWeb, CommonCrawl, and Gigaword.

In detail, the architecture of ELECTRA is characterized by 12 encoder layers, known as Transformers Blocks, and 12 attention heads (as introduced in [23], hence feed forward networks with a hidden size of 768. Each training session of the *c2f-coref* model has been fixed of 100 epochs, with a learning rate varying from 0.1 up to 0.00001. More architectural details and training hyper-parameters are reported in Table 2. At the top of Table, architectural details of ELECTRA are reported, whereas at the bottom, hyperparameters of *c2f-coref* model.

All experiments have been performed on a deep learning IBM cluster, composed by 13 operation nodes (AC922) characterized by 2 x 16 cores at 2.7 GHz, 4 x NVIDIA V100 GPUs (16GB), 512GB RAM, 100Gb IB EDR (2ports), 2 x 1.92TB SSD.

The choice of some values for the hyperparameters of both ELECTRA and *c2f-coref* model has been dictated by the hardware resource used for the experiments. In particular, *max_segment_length* has been set to 512 in order to maximize the length of each segment and, thus, the contextualized representation that can be obtained by using ELECTRA.

[1]https://github.com/lxucs/coref-hoi
[2]https://storage.googleapis.com/electra-data/electra_large.zip
[3]https://github.com/huggingface/transformers

TABLE 2. Hyper-parameters: the list is split into two parts, the top list contains ELECTRA hyperparameters, whereas the bottom list indicates *c2f-coref* model hyperparameters.

Hyperparameter	Value
Number of Attention Heads	12
Number of Hidden Layers	12
Hidden size	768
Number of Hidden Layers	12
Parameters	110M
Vocabulary Size	32102
Max Training Sentences	2
Max Segment Length	512
Max Span Width	40
Max Top Span Ratio	0.6
Max Top Antecedents	50
Epochs	100
Dropout	0.3
Learning rate	from 0.1 up to 0.00001
Loss	marginalized
FFNN size	1000
Feature Embedding size	20

Moreover, *max_training_sentences* has been fixed equal to 2 for computational limits, thus truncating documents with a length higher than 1024.

On the other hand, with reference to the parameters of *c2f-coref* model, *max_span_width* has been set equal to 40 since, according to what is shown in Figure 1, almost the totality of the mentions presents a width of at most 40 tokens. Moreover, *top_span_ratio* has been fixed to 0.6, since, according to what is reported in Figure 2, all partitions present a mentions-tokens ratio less or equal of 0.3. This value has been doubled in order to have a balanced number of positive and negative examples for the *c2f-coref* model.

B. EVALUATION METRICS

For the purpose of this work, official metrics provided by the Conll 2012 shared task have been taken into account. In particular, three metrics addressing different dimensions have been adopted: MUC, $B-CUBED$ and $CEAF_e$. These metrics consider the true set of entities K (named key or key partition) obtained through manual annotation of the entities, and the predicted (or response) set of entities R, i.e. answer partition produced by the system. In particular, they are defined as follows:

- MUC considers a cluster of references as linked references, i.e. each reference is linked to one or more references. It measures the number of link modifications required to make the result entity set R identical to the key entity set R. Precision is calculated as follows:

$$Precision_{\text{MUC}} = \frac{\sum_{r_j \in R} \frac{|r_j| - |P(r_j)|}{|r_j|}}{\sum_{r_j \in R}(|r_j| - 1)} \tag{13}$$

while recall is equal to:

$$Recall_{\text{MUC}} = \frac{\sum_{k_i \in K} \frac{|k_i| - |P(k_i)|}{|k_i|}}{\sum_{k_i \in K}(|k_i| - 1)} \tag{14}$$

Precision is calculated as the number of common pairs between entities in K divided by the number of pairs in R, while recall is equal to the number of common pairs between entities in K and R divided by the number of pairs in K.

- $B-CUBED$ (B^3) first computes precision and recall for each mention, and then calculates the weighted average of these individual precision and recall scores to obtain global precision and recall. In particular, for each mention m of K, the recall is computed by considering the fraction of the correct mentions included in the predicted entity that contains m. On the other hand, the precision is computed by exchanging the gold entities with the predicted ones. If K is the key entity containing mention m, and R is the response entity containing mention m, precision and recall for the mention m are calculated as:

$$Precision_{B^3} = \frac{\sum_{k_i \in K} \sum_{r_j \in R} \frac{|k_i \cap r_j|}{|k_i|}}{\sum_{r_j \in R} |r_j|} \tag{15}$$

$$Recall_{B^3} = \frac{\sum_{k_i \in K} \sum_{r_j \in R} \frac{|k_i \cap r_j|}{|k_i|}}{\sum_{k_i \in K} |k_i|} \tag{16}$$

- $CEAF_e$ uses a similarity measure to find the best one-to-one mapping between entities in K and entities in R. The best mapping is the one that maximizes the overall similarity of the entities, ϕ, which is given by the following equation:

$$\phi(k_i, r_i) = \frac{2|k_i \cap r_j|}{|k_i| + |r_i|} \tag{17}$$

Recall is the total similarity divided by the number of mentions in K:

$$Recall_{CEAF} = \frac{\sum_{k_i \in K^*} \phi(k_i, g^*(k_i))}{\sum_{k_i \in K} \phi(k_i, k_i)} \tag{18}$$

where g^* is a function associating to each entity of K an entity of R, whereas K^* is the set of key entities in the optimal mapping.
Precision is the total similarity divided by the number of mentions in R:

$$Precision_{CEAF} = \frac{\sum_{k_i \in K^*} \phi(k_i, g^*(k_i))}{\sum_{r_i \in R} \phi(r_i, r_i)} \tag{19}$$

VI. RESULTS AND DISCUSSION

In this section, experimental results are presented, analyzed and discussed from different perspectives, ether quantitatively, also reporting a comparison with the state of the art and deeply inspecting the typology of produced errors, and qualitatively, highlighting some examples of correctly and incorrectly predictions. Moreover, the contribution of each piece of the proposed model is inspected, analyzed and discussed by means of an ablation study.

A. QUANTITATIVE ANALYSIS

In Table 3 results obtained by most successful approaches proposed in the literature are shown. Results are reported with reference to the three metrics presented in V-B, according to the criteria proposed in Conll 2012 shared task. The table is divided into two parts: in the first one all the approaches based on the e2e-coref, c2f-coref and s2e-coref models are listed, including the proposed solution referred as *c2f-coref$_{opt}$ + ELECTRA-large* and highlighted in bold, whereas, in the second one, the approaches based on CorefQA model are mentioned.

The results shown in Table highlight that the proposed *c2f-coref$_{opt}$ + ELECTRA-large* model outperforms the best two models based on c2f-coref and s2e-coref architectures, namely *c2f-coref + SpanBERT-large* and *s2e-coref + Longformer-large*, with a F1 score boost of 1 and 0.9, respectively. These results also imply that ELECTRA and the way used to encode spans of text allows obtaining a more effective token and span representation, without requiring a specific NLM trained to handle either spans of text, as SpanBERT, or long documents, as Longformer.

On the other hand, the *c2f-coref$_{opt}$ + ELECTRA-large* model reaches inferior performance than *CorefQA + SpanBERT-large*, but without neither resorting to data augmentation to improve its generalization capability nor processing hundreds of individual context-question-answer instances for a single document, substantially worsening execution time, as reported by [13]. As further stated in [21], [24], this method has resulted very computationally expensive since it needs to run a transformer-based model to perform a different query on the same document many times. It also exhibiting some difficulties to scale to long documents.

B. ABLATION STUDY AND ANALYSIS

The performance of the proposed *c2f-coref$_{opt}$ + ELECTRA-large* model has been deeply inspected with respect to four main aspects that are different from the *c2f-coref + BERT-large* model: i) the method used to represent spans of text; ii) the number and length of textual segments given in input; iii) the number of spans of text to be processed as input; iv) the number of epochs to be used in the training process.

First of all, the behavior of the proposed *c2f-coref$_{opt}$ + ELECTRA-large* model has been assessed with respect to the strategy used to represent spans of text, by comparing the performance obtained in terms of average and standard deviation of F1 score on ten random tests, by concatenating the embedding representations of the boundary tokens either to the average of the embedding representations of all the span tokens, or to the soft heads of these spans, calculated as a weighted sum of the embedding representations of all the span tokens, as proposed in [11].

Figure 6 shows the results achieved with the two methods, with an average F1 score gain of 0.23 if the average strategy is preferred to the soft head one. It is worth noting that this average F1 score gain does not significantly exceed the

TABLE 3. Evaluation results on the English CoNLL-2012 shared task of different coreference models. The table is divided into two parts, the first part with the results of the models based on the (e2e, c2f, s2e) paradigms whereas the second part shows the CorefQA based method. The proposed method is ***c2f-coref$_{opt}$ + ELECTRA-large*** and it is highlighted in bold.

Model	MUC			B-CUBE			CEAF$_e$			Avg F1
	P	R	F1	P	R	F1	P	R	F1	
e2e-coref [11]	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
c2f-coref + ELMO [5]	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
e2e + BERT-large [7]	82.6	84.1	83.4	73.3	76.2	74.7	72.4	71.1	71.8	76.6
c2f-coref + BERT-large [8]	84.7	82.4	83.5	76.5	74.0	75.3	74.1	69.8	71.9	76.9
c2f-coref + SpanBERT-large [9]	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
c2f-coref + SpanBERT-large [10]	85.9	85.5	85.7	79.0	78.9	79.0	76.7	75.2	75.9	80.2
Simplified e2e-coref [21]	85.4	85.4	85.4	78.4	78.9	78.7	76.1	73.9	75	79.7
c2f-coref + Longformer-base [13]	85.0	85.0	85.0	77.8	77.8	77.8	75.6	74.2	74.9	79.2
c2f-coref + Longformer-large [13]	86.0	83.2	84.6	78.9	75.5	77.2	76.7	68.7	72.5	78.1
s2e-coref + Longformer-large [13]	86.5	85.1	85.8	80.3	77.9	79.1	76.8	75.4	76.1	80.3
c2f-coref$_{opt}$ + ELECTRA-large	87.0	86.4	86.7	80.9	79.0	79.9	75.9	78.3	77.1	81.2
CorefQA + SpanBERT-base [22]	85.2	87.4	86.3	78.7	76.5	77.6	76.0	75.6	75.8	79.9
CorefQA + SpanBERT-large [22]	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	83.1

deviation ranges. Therefore, the superiority of the average strategy cannot be decisively stated. However, the major simplicity of the average strategy with respect to the soft head one, which is a weighted average with weights to be learned during the training phase, and comparable, or even slightly better, the performance obtained experimentally, in fact, justify the preference for the former in the final *c2f-coref$_{opt}$ + ELECTRA-large* model.

Secondly, the *c2f-coref$_{opt}$ + ELECTRA-large* model has been evaluated by comparing the performance obtained in terms of average and standard deviation of F1 score on ten random tests, with respect to the number and length of textual segments it receives in input, varying the *max_training_sentence* and *max_segment_len* hyperparameters, and with respect to the number of spans to be selected for being processed, by modifying *max_span_width* and *max_top_span_ratio* hyperparameters.

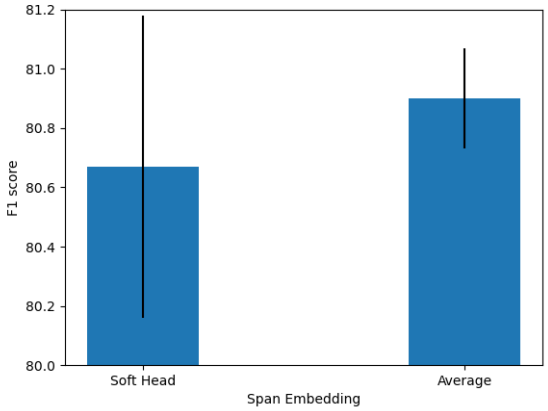
In particular, the hyperparameters *max_training_sentence* and *max_segment_len* are strictly correlated and indicates jointly the maximum number of tokens that can be managed by the model. More formally:

$$max\_token\_number = max\_training\_sentence * max\_segment\_len$$

The maximum number of tokens has an impact on the spatial complexity of the model. In the best result achieved by the model, it is equal to 1024, with *max_training_sentence* equal to 512 and *max_segment_len* equal to 2. However, in order to confirm this choice as the best one to improve the performance, other configurations for this couple of hyperparameters have been tested, varying *max_training_sentence* jointly, from 128 to 512, and *max_segment_len*, from 8 to 2, reaching a maximum number of tokens always equal to 1024 except for the couple (3-384) where it is equal to 1152.

The results in Figure 7 have shown that the usage of larger contexts always improves the performance.

Continuing to look at the Figure 7, two vertical bars are reported for each couple (*max_training_sentence*, *max_segment_len*), indicating the performance reached depending on the values they assume. The blue bar is

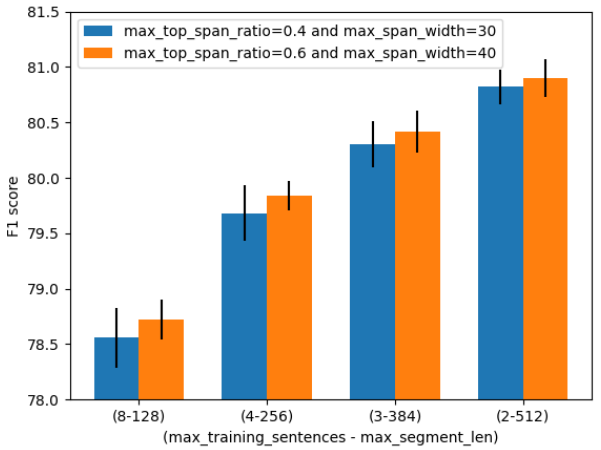


Soft Head	Average
80.67 ± 0.51	80.90 ± 0.17

FIGURE 6. The histogram and the related table show the ***c2f-coref$_{opt}$ + ELECTRA-large*** model behavior in terms of average and standard deviation of F1 score obtained on ten random tests. The results are function of the methods, namely average or soft head, used to combine token embeddings composing a span and create a span embedding.

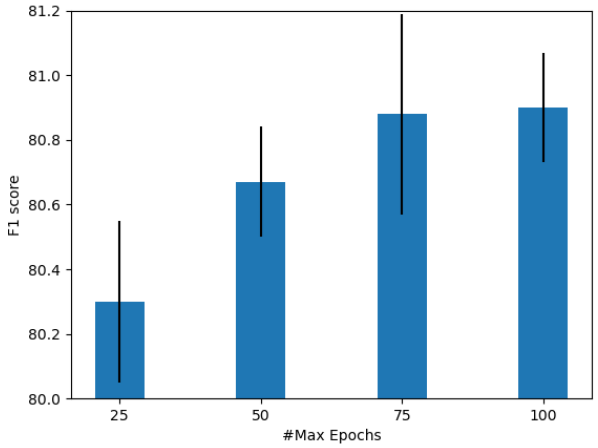
referred to a configuration with *max_top_span_ratio* equal to 0.4 and *max_span_width* equal to 30, whereas the orange bar to the same hyperparameters but with values equal to 0.6 and 40, respectively. It appears clear form the Figure that the couple (0.6, 40) for the *max_top_span_ratio* and *max_span_width* hyperparameters represents the best one, for each couple (*max_training_sentence*, *max_segment_len*) and, thus, regardless of it. Indeed, more spans are selected (0.6 and 40 for *max_top_span_ratio* and *max_span_width*), better performances the model is able to reach. Thus, the values (0.6, 40) for *max_top_span_ratio* and *max_span_width* and the values (2-512) for *max_training_sentence* and *max_segment_len* represent the best choice and have been all used to configure the final *c2f-coref$_{opt}$ + ELECTRA-large* model.

Finally, the number of epochs to be used for the training of the *c2f-coref$_{opt}$ + ELECTRA-large* model has been also assessed, varying it from 25 to 100 with an incremental step equal to 25. Figure 8 show the F1 score obtained in terms



max_top_span_ratio	max_span_width	(8-128)	(4-256)	(3-384)	(2-512)
0.6	40	78.72 ± 0.18	79.84 ± 0.13	80.42 ± 0.19	80.90 ± 0.17
0.4	30	78.56 ± 0.27	79.68 ± 0.25	80.30 ± 0.21	80.82 ± 0.16

FIGURE 7. The histogram and the related table show the *c2f-coref*$_{opt}$ + *ELECTRA-large* model behavior in terms of average and standard deviation of F1 score obtained on ten random tests. The results are function of the *max_training_sentence* and *max_segment_len* parameters. In detail, the blue bar is referred to a configuration with *max_top_span_ratio* equal to 0.4 and *max_span_width* equal to 30, whereas the orange bar to the same hyperparameters with values 0.6 and 40 respectively.



25	50	75	100
80.30 ± 0.25	80.67 ± 0.17	80.88 ± 0.31	80.90 ± 0.17

FIGURE 8. The histogram and the related table show the system behavior in terms of average and standard deviation of F1 score obtained on ten random tests. The results are function of the maximum number of epochs used during the training stage.

of average and standard deviation of F1 score on ten random tests.

It is worth noting that as the number of epochs increases, the model improves its performance, thus reaching the best value with 100 epochs, a greater value then the 20 epochs used by [8] and the 24 used by [10].

C. IMPACT ANALYSIS OF c2f-coref$_{opt}$ AND ELECTRA

First, the effectiveness of the proposed optimized c2f-coref$_{opt}$ architecture with ELECTRA in place of the original *c2f-coref* one equipped with BERT has been assessed. To this end, further experiments have been arranged and performed and the results achieved are reported in Table 4:

TABLE 4. Performance comparison among various models, varying the architecture used between the original *c2f-coref* and the proposed *c2f-coref*$_{opt}$ and the language model between BERT and ELECTRA in their large versions.

Model	Avg F1
c2f-coref + BERT-large	76.3
c2f-coref + ELECTRA-large	77.5
c2f-coref$_{opt}$ + BERT-large	77.9
c2f-coref$_{opt}$ + ELECTRA-large	81.2

The results obtained for the *c2f-coref* architecture equipped with BERT are slightly lower than the ones reported in [8]. Still, they have been experimentally obtained from scratch to use the same computing infrastructure for comparison.

As can be seen, the usage of ELECTRA in place of BERT in the *c2f-coref* architecture has generated an F1 score gain of 1.2, highlighting the superiority of the ELECTRA language model over the more widely used BERT model. For this test, the same hyperparameters used by the baseline proposed in [8] have been adopted, maintaining the best couple of values obtained for BERT for *max_training_sentence* and *max_segment_len*, i.e. 4 and 256, respectively.

Moreover, the proposed optimized *c2f-coref*$_{opt}$ architecture in place of the original one, maintaining BERT as the language model, has generated an F1 score gain of 1.6, highlighting the validity of the proposed optimizations, independently of the language model used. The optimized hyperparameters reported in Table 2 have been adopted for this test, but maintaining the best couple of values obtained in [8] for BERT for *max_training_sentence* and *max_segment_len*, i.e., 4 and 256, respectively.

Finally, the contextual usage of ELECTRA in place of BERT and the proposed *c2f-coref*$_{opt}$ architecture in place of the original one, has generated an F1 score gain of 4.9, highlighting the goodness of both. To achieve the result of this last model, the optimized hyperparameters reported in Table 2 have been adopted, choosing to use the best couple of values obtained for ELECTRA for *max_training_sentence* and *max_segment_len*, i.e., 2 and 512, respectively.

As a second analysis, the different behavior of the *c2f-coref*$_{opt}$ + *ELECTRA-large* model has been evaluated in comparison with the one reported by [8] for the *c2f-coref* + *BERT-large* model, when various configurations of values are considered for *max_training_sentence* and *max_segment_len* hyperparameters.

In particular, in Table 5, the performance achieved by the *c2f-coref* + *BERT-large* model proposed in [8] has been compared with those obtained by first changing the architecture with the proposed *c2f-coref*$_{opt}$ and then the language model with ELECTRA.

As highlighted by the results, both the models using BERT have performed very similarly or worse when *max_segment_len* of 384 or 512 is used, reaching the worst

TABLE 5. Performance comparison among different models, varying values for *max_training_sentence* and *max_segment_len* hyperparameters. The results in the first line come from [8] and are compared with the ones achieved first by changing the architecture with the proposed *c2f-coref$_{opt}$* and then the language model with ELECTRA in its large version. The baseline results are reported only as mean values, whereas the novel computed ones as mean ± standard deviation.

Model	max_training_sentence - max_segment_len			
	$8-128$	$4-256$	$3-384$	$2-512$
c2f-coref + BERT-large	76.6 [8]	76.9 [8]	77.3 [8]	73.6 [8]
c2f-coref$_{opt}$ + BERT-large	77.12 ± 0.19	77.70 ± 0.17	77.55 ± 0.01	73.83 ± 0.27
c2f-coref$_{opt}$ + ELECTRA-large	78.72 ± 0.18	79.84 ± 0.13	80.42 ± 0.19	80.90 ± 0.17

F1 score with the couple (2-512). On the other hand, the proposed *c2f-coref$_{opt}$ + ELECTRA-large* model has increasingly taken advantage of using greater *max_segment_len* with the best F1 score achieved for the couple (2-512).

A possible motivation of this difference could be the usage in ELECTRA, similarly to SpanBERT, of a single-sequence training, that has been shown in [9] to work better than bi-sequence training with next sentence prediction objective used by BERT. Indeed, like SpanBERT, ELECTRA has removed the next sentence prediction objective and the two-segment sampling procedure used by BERT for it, performing the training on single contiguous segments instead of two half-length segments. This removal may have produced a better handling of longer full-length contexts, allowing to more proficiently learn longer-range features.

D. ERROR ANALYSIS

A further error analysis has been carried out in order to assess the performance of *c2f-coref$_{opt}$ + ELECTRA-large* model with respect to the prediction of mentions depending on their length and, in case of single-token ones, depending on the different Parts-Of-Speech (POS) they belong to.

In particular, Table 6 outlines the numbers of mentions that are predicted (*p*) or not correctly predicted (*np*), grouped in two classes, namely *single-token* and *multi-token*. With respect to the *multi-token* class, it has been further subdivided in three sub-classes depending on their token length.

TABLE 6. Predicted and not-predicted mentions, grouped depending on their length.

Mention Class	Total	Predicted	Not-Predicted
Single-Token	11215	10404 (92.8%)	811 (7.2%)
Multi-Token	8548	7245 (84.8%)	1303 (15.2%)
$\longrightarrow (2-9)$ Tokens	7721	6629 (85.9%)	1092 (14.1%)
$\longrightarrow (10-19)$ Tokens	635	500 (78.8%)	135 (21.2%)
$\longrightarrow (>20)$ Tokens	192	127 (66.1%)	65 (33.9%)

It is possible to highlight a greater number of correctly predicted mentions in case they are single-token (92.8%) than in case they are multi-token (84.8%). Moreover, for multi-token mentions, also a progressive decay in the correct predictions is shown as the number of tokens constituting the mentions increases. The percentage of correctly predicted multi-token mentions, indeed, decreases from 85.9% for mentions between 2 and 9 in length to 66.1% for mentions greater than 20 in length.

Going deeply with respect to single-token mentions, Table 7 outlines the numbers of this type of mentions that are predicted (*p*) or not correctly predicted (*np*), grouped with reference to the three most frequent POS categories, namely pronoun (PRON), proper noun (NOUN) and verb (VERB). In the PRON category, personal, possessive and demonstrative pronouns have been considered, where, in the last two categories, also adjectives have been included since POS-tagged in the same way in the dataset.

The second column shows the total number of occurrence for every POS with the frequencies in round brackets. As confirmed by other studies, the most used single-token POS to co-refer an entity is pronoun, which covers 72.9% of total single-token mentions.

TABLE 7. Predicted and not predicted single-token mentions grouped according to the most frequent POS classes.

POS	Total	Predicted	Non-Predicted
PRON	$8611\ (72,9\%)$	$8283\ (96.2\%)$	$328\ (3.8\%)$
personal	$6420\ (78,5\%)$	$6232\ (97.1\%)$	$188\ (2.9\%)$
possessive	$1758\ (21,5\%)$	$1726\ (98.2\%)$	$32\ (1.8\%)$
demonstrative	$433\ (5\%)$	$325\ (75.1\%)$	$108\ (24.9\%)$
NOUN	$2190\ (19,5\%)$	$1888\ (86.2\%)$	$302\ (13.8\%)$
VERB	$325\ (2,9\%)$	$220\ (67.7\%)$	$105\ (32.3\%)$

Concerning the mentions belonging to pronoun POS, the proposed *c2f-coref$_{opt}$ + ELECTRA-large* model reaches a percentage of wrong predictions equal to 3.8%. The error rate for mentions associated to the noun POS is slightly higher reaching a value of $13,8\%$. With regard to verb POS, the model has an error rate of nearly 32.3%, but the impact on the overall results is however low due to the fact that verbs only affect $2,9\%$ of the total POS categories associated to single-token mentions in the dataset.

A deeper analysis of mentions associated to pronoun POS shows differences between the predicted and erroneous pronoun types (as shown in Table 7). Almost 80% of pronouns used as mentions are personal pronouns, which are predicted with an accuracy of 97.1%. Correct predictions increase by just over one percentage point (98.2%) in the case of possessives, which account for 21.5% of the total number of pronouns. By contrast, mention predictions are much more inaccurate in case of demonstratives: the model stops at 75.1% of correct predictions, while the error rate increases by around 25%.

E. QUALITATIVE ANALYSIS

A qualitative analysis has been also carried out in order to highlight clusters that are both correctly and incorrectly

TABLE 8. Examples of correctly (bold) and incorrectly (italicized) predicted clusters with reference to both single-token mentions, grouped depending on the most frequent POS categories, and multi-token mentions, grouped depending on their length.

No. of Tokens	Sub-Type	Cluster Prediction	Examples
13*Single	3*PRON	Correct	You would not care about the ballistic missile program if they did not have **the nuclear weapons program.** [...] We had to be very firm about **it**.
		Incorrect	To express *its* determination , the *Chinese* securities regulatory department compares this stock reform to a die that has been cast .
	3*NOUN	Correct	[...] in two thousand when **George Bush** became president [...] although the **Bush** administrat's rhetoric about Kim Jong and his regime has sometimes seemed ferocious North Korea's leaders
		Incorrect	*The oil industry*'s middling profits could persist through the rest of the year [...]*industry* executives and analysts say reducing chances
	3*VERB	Correct	[...] application to **buy** five Valley Federal branches [...] **The broken purchases** appear as additional evidence of trouble [...]
		Incorrect	[...] to *bomb* an American ship in Yemen. [...] he also provided details of a plan to *attack* an American warship [...]
13*Multi	3*(2 - 9)	Correct	[...] Well, ah, **Professor Zhou**, as this incident involved many departments [...] [...] how do **you** think the various department [...]
		Incorrect	Now *your new book The world is Flat* I've got *it* here. [...] uh *monster best seller* [...]
	3*(10 - 19)	Correct	[...]Next is **Yang Yang , a host of Beijing Traffic Radio Station** .[...] [...] And **you**, **Yang Yang** ? [...]
		Incorrect	Let me show you all *two presidents Bill Clinton back in nineteen ninety three George Bush ten years* later talking to the North Koreans in effect . [...] [...] *President Clinton President Bush* [...]
	3*(>20)	Correct	Our correspondent specially interviewed **Director Meng Xianlong, Mr. Meng Xianlong, of the Command Center of the Beijing Municipal Traffic Administration**. [...] Let's hear what **Director Meng** said [...]
		Incorrect	[...] In recent years, *Reader 's Digest, New York Times Co.'s McCall 's, and most recently News Corp. 's TV Guide,* have cut *their* massive circulation rate bases [...]

predicted by the proposed *c2f-coref*$_{opt}$ + *ELECTRA-large* model.

In detail, in Table 8, examples of correctly and incorrectly predicted clusters are reported. These examples are first arranged depending on the length of the mentions composing the clusters, namely single-token and multi-token. For the single-token class, the clusters highlighted in the examples contain at least one mention whose length is equal to 1. Moreover, for this typology, a further subdivision is foreseen with respect to pronoun, noun and verb POS categories. For each of these POS categories, every cluster reported in the examples contains at least one single-token mention associated to that POS. On the other hand, for the multi-token class, the clusters in the examples contain at least one mention whose length is included in the ranges (2-9), (10-19) or (>20), respectively.

Moreover, with reference to the examples in the last column, mentions belonging to correctly predicted clusters are reported as **boldfaced**, whereas the ones belonging to incorrectly predicted clusters are written as underlined. In order to make examples more readable, their text is truncated by indicating ellipsis in square brackets.

Concerning clusters containing at least a single-token mention, for the pronoun POS category, the model correctly predicts the neuter third-person personal pronoun ''it'' in prepositional phrase postponed to the verb (first example in table 8), making a cluster with the multi-token mention ''the nuclear weapons program''.

By contrast, an error is shown in the second example, where in the subordinate clause ''To express its determination'', the possessive ''its'', is not correctly predicted, and thus not properly associated to the other mention expressed by the proper noun ''Chinese'' in the main clause ''the Chinese securities regulatory department compares this stock reform to a die that has been cast''. This behavior is probably due to the fact that the second example has a more complex syntactic construction made of a main clause and a subordinate one.

Observing the examples concerning single-token mentions belonging to the noun POS category, it can be seen that the proper noun ''Bush'' is correctly predicted as mention and linked to the mention ''George Bush'' to form a cluster. On the other hand, in the second example, the common noun ''industry'' is not predicted as mention. This occurs despite the fact that the correctly predicted mention ''Bush'' in the first example is in an apparently more complex syntactic construction, acting as a subject within a subordinate clause introduced by the preposition ''although''. By contrast, for the not predicted mention ''industry'', the second example presents a linear easier syntax ''industry executives…''. It is thus possible to assume for the noun POS category - which overall success rate is more than 86% - that proper names have a greater effect over common ones. In the absence of sub-tags on the type of nouns, this behavior can only be hypothesized.

With regard to the couple of examples concerning single-token mentions that are verbs, there is a slightly different behavior. No specific factor, such as mention position or syntactic construction, seems to influence whether or not the mentions are predicted and clustered correctly. It is therefore not possible to identify a clear-cut discrimination between correct and incorrect predictions. However, it should be noted - as mentioned above - that verbs are the POS category with the highest percentage of error (49.23%) but that they account for very little of the total grammatical categories (2.89%).

Concerning clusters containing at least a multi-token mention, although the examples associated to the different token intervals are very heterogeneous, a recurring pattern can be noted. Correctly predicted clusters contain multi-token mentions tending to be proper nouns composed by two or more tokens, as in the case of ''Professor Zhou'' for mentions within $(2-9)$ interval, or noun phrases with topicalizations. The proper noun acting as subject (for instance ''Yang Yang'' in $(10-19)$ interval is located on the left of the clause that refers to it (''a host of Beijing Traffic Radio Station''), instead of the canonical order. This phenomenon can be observed for every class with clauses referring to the subject that becomes progressively longer. By contrast, incorrectly predicted clusters contain mentions presenting a plain subject-verb-object (SVO) syntax (i.e. ''your new book…I've got here…'') but with several proper nouns within paratactic conjunctions, as in ''president Bill Clinton back…George Bush'', or punctuation marks ''Reader's Digest, New York Times…''. These constructions could be the cause of a higher difficulty in correctly determining mentions and clusters containing them.

VII. CONCLUSION

In this paper, the most used end-to-end coreference architecture proposed in [5], [11] has been enhanced adopting ELECTRA as representation layer in order to encode input tokens and improve their contextualized representation.

Moreover, the architecture itself has been revised: i) by representing spans of text considering both the context they appear, exploiting information at their boundary tokens, and their entire content, averaging information of all their tokens, ii) by optimizing both number and length of textual segments considered in input in order to preserve the computational efficiency, maximize the coverage of the input documents and better exploit the representation power of ELECTRA, iii) by maximizing the number of spans of text to be processed, since potentially representing mentions, not violating computational limits due a high spatial complexity.

The final model has been assessed on the OntoNotes dataset from both a quantitative and qualitative perspective, showing its effectiveness with respect to the state-of-the-art. In detail, the model has outperformed all the other existing solutions derived from the architecture proposed in [5], [11], whereas it has reached inferior performance than [22], but without neither resorting to data augmentation to improve its generalization ability nor requiring a major execution time to process a plenty of individual context-question-answer instances for each single document.

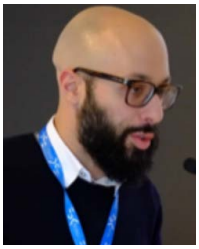
These results have also implied that ELECTRA and the way used to encode spans of text allows obtaining a more effective token and span representation. These aspects leave open the possibility of using this model also for low-resource languages for which it is sufficient that a pre-trained version of ELECTRA exists, not requiring a language-specific NLM trained to handle either spans of text, as SpanBERT, or long documents, as Longformer.

REFERENCES

[1] R. Sukthanker, S. Poria, E. Cambria, and R. Thirunavukarasu, "Anaphora and coreference resolution: A review," *Inf. Fusion*, vol. 59, pp. 139–162, Jul. 2020.
[2] N. Stylianou and I. Vlahavas, "A neural entity coreference resolution review," *Exp. Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114466.
[3] D. Ji, J. Gao, H. Fei, C. Teng, and Y. Ren, "A deep neural network model for speakers coreference resolution in legal texts," *Inf. Process. Manage.*, vol. 57, Aug. 2020, Art. no. 102365.
[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Minneapolis, MN, USA, vol. 1, Jun. 2019, pp. 4171–4186.
[5] K. Lee, L. He, and L. Zettlemoyer, "Higher-order coreference resolution with coarse-to-fine inference," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, New Orleans, LO, USA, vol. 2, Jun. 2018, pp. 687–692.
[6] H. Fei, X. Li, D. Li, and P. Li, "End-to-end deep reinforcement learning based coreference resolution," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, Jul. 2019, pp. 660–665.
[7] B. Kantor and A. Globerson, "Coreference resolution with entity equalization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, Jul. 2019, pp. 673–677.
[8] M. Joshi, O. Levy, L. Zettlemoyer, and D. Weld, "BERT for coreference resolution: Baselines and analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong, Nov. 2019, pp. 5803–5808.
[9] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 64–77, Jan. 2020.
[10] L. Xu and J. D. Choi, "Revealing the myth of higher-order inference in coreference resolution," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Nov. 2020, pp. 8527–8533.
[11] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, Sep. 2017, pp. 188–197.
[12] I. Beltagy, E. Matthew Peters, and A. Cohan, "Longformer: The long-document transformer," *CoRR*, vol. abs/2004.05150, pp. 1–17, Apr. 2020.
[13] Y. Kirstain, O. Ram, and O. Levy, "Coreference resolution without span representations," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, Aug. 2021, pp. 14–19.
[14] K. Clark, M.-T. Luong, V. Q. Le, and D. C. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*. Addis Ababa, Ethiopia, Apr. 2020, pp. 1–18.
[15] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how BERT works," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 842–866, Dec. 2020.
[16] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes," in *Proc. Joint Conf. EMNLP CoNLL-Shared Task*, Jeju Island, South Korea, Jul. 2012, pp. 1–40.
[17] S. Wiseman, A. M. Rush, S. Shieber, and J. Weston, "Learning anaphoricity and antecedent ranking features for coreference resolution," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, Beijing, China, 2015, Jul. 2015, pp. 1416–1426.
[18] S. Wiseman, A. M. Rush, and S. M. Shieber, "Learning global features for coreference resolution," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, San Diego, CA, USA, Jun. 2016, pp. 994–1004.
[19] K. Clark and C. D. Manning, "Deep reinforcement learning for mention-ranking coreference models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, Nov. 2016, pp. 2256–2262.
[20] K. Clark and D. C. Manning, "Improving coreference resolution by learning entity-level distributed representations," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, vol. 1, Aug. 2016, pp. 643–653.
[21] T. M. Lai, T. Bui, and D. S. Kim, "End-to-end neural coreference resolution revisited: A simple yet effective baseline," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 8147–8151, doi: 10.1109/ICASSP43922.2022.9746254.
[22] W. Wu, F. Wang, A. Yuan, F. Wu, and J. Li, "CorefQA: Coreference resolution as query-based span prediction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 6953–6963.
[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon *et al.*, Eds. Long Beach, CA, USA: Curran Associates, Dec. 2017, pp. 5998–6008.
[24] R. Thirukovalluru, N. Monath, K. Shridhar, M. Zaheer, M. Sachan, and A. McCallum, "Scaling within document coreference to long texts," in *Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP)*, 2021, pp. 3921–3931.

FRANCESCO GARGIULO received the M.Sc. degree *(cum laude)* in telecommunication engineering and the Ph.D. degree in information and automatic engineering from the University of Naples Federico II, in 2006 and 2009, respectively. He is currently a Technological Researcher with the Institute for High Performance Computing and Networking, National Research Council, Italy (ICAR-CNR). He has been involved in different national and European projects. He has authored numerous peer-reviewed articles on international journals and conference proceedings. His current research interests include e-health, big data analytics, natural language processing, artificial intelligence, and deep learning. He has been on the program committee of international conferences and workshops and, moreover, is also a member of the editorial board of some international journals.

ANIELLO MINUTOLO received the M.Sc. degree in computer science engineering from the University of Naples Federico II, and the Ph.D. degree in information technology engineering from the University of Naples Parthenope. Since 2018, he has been a Contract Professor of informatics at the Faculty of Engineering, University of Naples Federico II. He is currently a Researcher at the Institute for High Performance Computing and Networking, National Research Council of Italy (ICAR-CNR). His current research interests include artificial intelligence, decision support systems, dialog systems, and knowledge management, modeling, and reasoning. He has been involved in different national and European projects, he has been on the program committee of some international conferences and workshops and, moreover, is also a member of the editorial board of some international journals.



RAFFAELE GUARASCI received the M.Sc. degree in computational linguistics from the University of Pisa, and the Ph.D. degree in computational linguistics from the University of Salerno. He is currently a Researcher at the Institute for High Performance Computing and Networking of the National Research Council of Italy (ICAR-CNR). He is a member of the ''Cognitive Systems'' Laboratory, CNR-ICAR. He has been an Adjunct Professor with the Ph.D. Program in Computational Linguistics & Social Media, University of Salerno, and he has been on the program committee of national and international conferences. His current research interests include interaction between neural language models and linguistics theoretical and cognitive issues to the application of artificial intelligence methods based on machine/deep learning to natural language processing tasks.

EMANUELE DAMIANO received the M.Sc. degree in computer science from the University of Naples Federico II. He is currently a Graduate Research Fellow at the Institute for High Performance Computing and Networking of the National Research Council of Italy (ICAR-CNR). He is a member of the ''Cognitive Systems'' Laboratory, CNR-ICAR, and been involved in different national projects. His research interests include artificial intelligence approaches based on machine/deep learning applied to natural language processing and question answering.

GIUSEPPE DE PIETRO is currently the Director of the Institute for High Performance Computing and Networking, CNR, and an Adjunct Professor with the College of Science and Technology, Temple University, Philadelphia. He has been actively involved in many European and national projects, with industrial co-operations. He is the author of over 200 scientific papers published in international journals and conferences. His current research interests include cognitive computing, clinical decision support systems, and software architectures for e-health. He is also a KES International Member. He is also involved in many program committees and journal editorial boards.

HAMIDO FUJITA (Senior Member, IEEE) received the B.S. degree in electrical engineering from The University of Manchester, Manchester, U.K., in 1979, the master's and Ph.D. degrees in information engineering from Tohoku University, Sendai, Japan, in 1985 and 1988, respectively, and the Doctor Honoris Causa degree from Óbuda University, Budapest, Hungary, in 2013, and the Doctor Honoris Causa degree from Timisoara Technical University, Timisoara, Romania, in 2018. He received the title of an Honorary Professor from Óbuda University, in 2011. He is currently a Professor of artificial intelligence with Iwate Prefectural University, Takizawa, Japan, and as the Director of Intelligent Software Systems. He is an Adjunct Professor of computer science and artificial intelligence with Stockholm University, Stockholm, Sweden; University of Technology Sydney, Ultimo, NSW, Australia; National Taiwan Ocean University, Keelung, Taiwan; and others. He has supervised Ph.D. students jointly with the University of Laval, Quebec City, QC, Canada; University of Technology Sydney; Oregon State University, Corvallis, OR, USA; University of Paris 1 Pantheon-Sorbonne, Paris, France; and University of Genoa, Genoa, Italy. He has four international patents in software system and several research projects with Japanese industry and partners. He has given many keynotes in many prestigious international conferences on intelligent system and subjective intelligence. He headed a number of projects, including intelligent HCI, a project related to mental cloning for healthcare system as an intelligent user interface between human users and computers, and SCOPE project on virtual doctor systems for medical applications. He was a recipient of the Honorary Scholar Award from the University of Technology Sydney in 2012. He is the Editor-in-Chief of *Knowledge-Based Systems*. He is the Vice President of the International Society of Applied Intelligence, and also the Editor-in-Chief of *Applied Intelligence* (Springer). He is also Highly Cited Researcher in Cross-field for the year 2019 and Computer Science for the year 2020 by Clarivate Analytics.

MASSIMO ESPOSITO received the M.Sc. degree *(cum laude)* in computer science engineering from University of Naples Federico II, in March 2004, the University 1st Level master's degree, named European Master on critical networked systems, in December 2007, and the Ph.D. degree in information technology engineering from the University of Naples Parthenope, in April 2011. Since 2012, he has been a Contract Professor of informatics at the Faculty of Engineering, University of Naples Federico II. Since 2016, he has been responsible of the Laboratory ''Cognitive Systems,'' Institute for High Performance Computing and Networking of the National Research Council of Italy (ICAR-CNR). He is currently a Researcher at the ICAR-CNR. He is the author of over 100 peer-reviewed papers on international journals and conference proceedings. His current research interests include artificial intelligence (AI) and are focused on AI algorithms and techniques, mixing deep learning and knowledge-based technologies, for building intelligent systems able to converse, understand natural language and answer to questions, with emphasis on the distributional neural representation of words and sentences, and on specific natural language tasks, such as part of speech tagging, sentence classification, and open information extraction. He has been involved in different national and European projects, he has been on the program committee of many international conferences and workshops and, moreover, is also a member of the editorial board of some international journals.