Computers in Biology and Medicine 152 (2023) 106415

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Machine understanding surgical actions from intervention procedure textbooks

Marco Bombieri [a,*], Marco Rospocher [b], Simone Paolo Ponzetto [c], Paolo Fiorini [a]

[a] *Department of Computer Science, University of Verona, Verona, Italy*
[b] *Department of Foreign Languages and Literatures, University of Verona, Verona, Italy*
[c] *Data and Web Science Group, University of Mannheim, Mannheim, Germany*

ARTICLE INFO

Keywords:
Semantic role labeling
Surgical data science
Procedural knowledge
Information extraction
Natural language processing

ABSTRACT

The automatic extraction of procedural surgical knowledge from surgery manuals, academic papers or other high-quality textual resources, is of the utmost importance to develop knowledge-based clinical decision support systems, to automatically execute some procedure's step or to summarize the procedural information, spread throughout the texts, in a structured form usable as a study resource by medical students. In this work, we propose a first benchmark on extracting detailed surgical actions from available intervention procedure textbooks and papers. We frame the problem as a Semantic Role Labeling task. Exploiting a manually annotated dataset, we apply different Transformer-based information extraction methods. Starting from RoBERTa and BioMedRoBERTa pre-trained language models, we first investigate a zero-shot scenario and compare the obtained results with a full fine-tuning setting. We then introduce a new ad-hoc surgical language model, named SurgicBERTa, pre-trained on a large collection of surgical materials, and we compare it with the previous ones. In the assessment, we explore different dataset splits (one in-domain and two out-of-domain) and we investigate also the effectiveness of the approach in a few-shot learning scenario. Performance is evaluated on three correlated sub-tasks: predicate disambiguation, semantic argument disambiguation and predicate-argument disambiguation. Results show that the fine-tuning of a pre-trained domain-specific language model achieves the highest performance on all splits and on all sub-tasks. All models are publicly released.

1. Introduction

Thousands of surgeries are performed every day in hospitals around the world. Surgery is a complex profession that requires years of study and practice to master. Usually, the course of study of a surgeon consists of the first part of theoretical study, in which the surgeon acquires the fundamental theoretical notions of the profession, and a final part of practice, in which the student, through a cycle of internships, integrates the theoretical knowledge learned with experience. Since theoretical study occupies a predominant and substantial part of the study cycle of an apprentice surgeon, the literature is teeming with manuals, online resources and academic papers of the highest quality, used by universities around the world. These texts usually contain two different types of information [1]:

- *procedural knowledge*, the one possessed by an intelligent agent (in surgery, a surgeon or a surgical robot) able to perform a task (a surgical intervention). Typically, the description of a procedure details a set of surgical actions linked together temporally and causally;
- *non-procedural knowledge*, the one that does not express an action executable by an intelligent actor, but rather additional, related knowledge, for instance about anatomy.

This large amount of high-quality procedural information, if automatically processed by Natural Language Processing (NLP) techniques, is valuable content that could be exploited in many clinical applications. For example, robots could automatically build or extend a proper surgical knowledge-base, reasoning with it in realistic intervention scenarios. Humans could benefit from it for question answering applications, usable for example in an early learning phase by medical students. However, so far the extraction of surgical knowledge directly from surgery manuals and textbooks has received little attention from the scientific community, as current trends mostly focus on the derivation of knowledge from kinematic and video data captured by endoscopic sensors and cameras during interventions [2,3], or on the manual modeling of ontologies [4]. In this paper, we tackle the yet unexplored problem of extracting procedural knowledge from textual surgical resources.

[*] Department of Computer Science, University of Verona, Verona, Italy.
E-mail address: marco.bombieri_01@univr.it (M. Bombieri).

https://doi.org/10.1016/j.compbiomed.2022.106415
Received 24 September 2022; Received in revised form 23 November 2022; Accepted 4 December 2022
Available online 6 December 2022

A procedure is an ordered sequence of actions linked together temporally and causally. An action may be activated when a certain pre-condition is satisfied and it reaches its end state when a certain post-condition occurs. An action, expressed in surgery with verbs (e.g., "dissect") or nominalized verbs (e.g., "dissection") [5], is accompanied by a set of semantic information, such as the *"agent"*, i.e., the one who performs the action; the *"patient"*, i.e., the one who undergoes the action; the *"instrument"*, which refers to the tool used for performing the action, and the *"purpose"* describing the reason why the action is performed. In addition, other semantic information is *temporal* and *spatial* parameters. This work specifically tackles the automatic extraction of the actions and aforementioned related semantic information from procedure intervention descriptions available in real-world robotic-surgery textbooks, i.e., the concrete resources used by apprentice surgeons to learn the interventions, contributing to advance the state-of-the-art of the field in several ways.

First, we propose to frame the extraction of procedural actions and related information from surgical texts as a Semantic Role Labeling (SRL) problem [6]. SRL is the well-established NLP task of labeling semantic arguments of predicates in sentences to identify "Who" does "What" to "Whom", "How", "When" and "Where". By framing the problem as a SRL task, predicates (verbs and nominalized verbs) in a sentence denotes some procedural actions, while the semantic arguments of the predicates indicate the actions' related semantic information. Since surgery is an unexplored and less-resourced domain for SRL, which, in line with most work in NLP, has focused in the past primarily on newswire text [7], we first investigate the ability of state-of-the-art language models trained on general-English (RoBERTa [8]) or biomedical (BioMedRoBERTa [9]) annotated texts to cope with this very specific domain. Then, we investigate how to improve the extraction quality by fine-tuning existing models on SRL-labeled, manually annotated, domain text, i.e., procedural sentences where surgical actions and related semantic information are accurately identified and tagged by users knowledgeable of the domain. Since manual annotation is very expensive and requires domain experts, we furthermore adopt unsupervised domain adaptation techniques — namely injecting a large quantity of unlabeled domain text into the models to augment their understanding of surgical language knowledge — to verify if results can be further improved, contributing a new language model (SurgicBERTa) specifically for the surgical domain. We compare all the considered and contributed models in an extensive quantitative evaluation, concretely investigating the following research questions:

- RQ1: How well are available general-English and bio-medical pre-trained language models able to perform SRL on surgical annotated texts without resorting to supervised learning (i.e., zero-shot learning)?
- RQ2: Does fine-tuning on surgical annotated texts substantially improve the performance with respect to the zero-shot setting using off-the-shelf models available in the literature?
- RQ3: How many annotated data are needed to attain substantial improvements via supervised learning for this task (i.e., few-shot learning)?
- RQ4: Does further unsupervised learning of pre-trained language models (as in our novel language model, named SurgicBERTa) help to better understand surgical language?
- RQ5: Are the SRL models able to generalize over different surgical sub-domains?

Besides exploiting the standard evaluation measures for the SRL task, we also propose a new way for evaluating SRL systems, based on the joint disambiguation of arguments and predicates, i.e., on the correct disambiguation of semantic arguments with respect to the correct meaning (i.e., sense) of the actual predicate. Finally, we publicly release[1] all the trained models and evaluation materials, contributing the research community a wealth of resources to support further research and development activities on the topic.

To the best of our knowledge, this work is the first one dealing with automatic procedural knowledge extraction from available robotic-surgery textbooks.

The paper is organized as follows. In Section 2, we revise some state-of-the-art works about the NLP methods used by the bio-medical community, applications of NLP in medicine and surgery, and techniques for procedural knowledge understanding of texts in different domains. In Section 3, we describe the SRL task, the neural-network architecture used, the annotated data exploited for training, validation and testing, the pre-trained language models considered, the evaluation techniques and some computational data. In Section 4, we present and discuss the obtained results. We conclude with Section 5, summarizing the main contributions of this paper and addressing related research directions for future works.

2. Background

While the field of biomedical NLP has a long history – see, among others, [10] for an overview and the proceedings of the long-standing ACL Workshop on Biomedical Language Processing [11] for up-to-date contributions – to the best of our knowledge, no works have tackled so far the problem of extracting procedural knowledge from surgical books or academic papers. Nevertheless, the literature includes various approaches for extracting relevant information from medical or surgical operative notes using NLP or extracting procedural information from other non-surgical domains. Consequently, this section overviews relevant previous works in three different related areas: the first part summarizes NLP methods traditionally used for bio-medical free-text mining; the second part discusses recent relevant applications of NLP techniques to the bio-medical and surgical domains; the third part presents papers dealing with the extraction of procedural knowledge from texts, considering also domains other than the bio-medical one.

NLP methods for the bio-medical domain. This paragraph summarizes the main NLP methods used for medical free-text mining. Among NLP techniques, those based on deep learning methods (i.e., neural networks) have become extremely popular also in medical NLP because of their higher performance on a plethora of tasks and no need for handcrafted features [12]. When applied to textual content, neural networks are typically fed with word embeddings, that is numerical representations of textual tokens. The most used word embeddings in the medical community in the last years were word2vec [13], Glove [14], and fastText [15]. These word embeddings have the potential to capture semantic relationships between pairs of words and/or syntactic information from unstructured text data. Furthermore, fastText is traditionally used for its subwords mechanism able to deal with out-of-vocabulary problems, very frequent when working with medical terminology. More recent medical language models associate each word with every other word in a sentence thanks to the bi-directional self-attention mechanism, outperforming previous baselines in many biomedical tasks [9,16,17]. The models presented in Sections 3.5.1 and 3.5.2 belong to this last category.

A critical element in bio-medical NLP is to have an available dataset to train and validate models. Published papers often used private datasets which are rarely shared, mostly due to patient privacy concerns [12], hindering the replication of the results. The most popular datasets and databases in the bio-medical NLP are MIMIC [18], the ones from i2b2 challenges (e.g., [19] for concept extraction) and the one from SemEval challenges (e.g., [20] for temporal relations extraction from clinical narratives). Unfortunately, none of them contains annotations of procedural surgical descriptions for semantic information extraction.

[1] https://gitlab.com/altairLab/surgical_srl

Application of NLP techniques to bio-medical domains. This paragraph summarizes some recent and relevant applications of NLP techniques to bio-medical and surgical domains. In [21], the authors use logistic regression with unigrams and concept unique identifiers from the unified medical language system to automatically predict the severity of chest injury after trauma from clinical notes. [22] proposes rule-based NLP algorithms to automatically extract surgery-specific data elements (category of knee arthroplasty, laterality, constraint type, whether patella resurfacing was performed or not, and implant model numbers) from knee arthroplasty operative notes: the main objective was to decrease the need for costly manual chart review and to improve data quality using NLP techniques. In [23], they use information extraction techniques applied to operative notes to detect the presence of variables associated with periprosthetic joint infection, including the growth of cultured organisms, documentation of inflammation, presence of sinus tract, and purulence. In [24], the authors use an extreme gradient boosting NLP machine learning algorithm [25] for automated detection of incidental durotomies in free-text operative notes of patients undergoing lumbar spine surgery. The clinical goal is to automatically survey the incidental durotomy that could be potential implications for postoperative recovery, patient-reported outcomes, length of stay, and costs. In [26] the authors address the detection of procedural knowledge in MEDLINE abstracts. In their work, procedural knowledge is defined as a set of *unit procedures* (each consisting of a *Target*, *Action* and *Method*) organized for solving a specific purpose. The proposed solution works in two steps. First, support vector machines and conditional random fields are combined for detecting sentences (purpose/solution) that may contain unit procedures, feeding them with content (unigrams and bigrams), position (sentence number in the abstract), neighbor (content features of nearby sentences) and ontological features (usage of terms from reference vocabularies). Then, sequence labeling with CRFs is performed to identify the components of unit procedures. In [27] the authors propose an NLP approach to automatically label right ventricular dysfunction size and function [28] from echocardiographic free text reports. In particular, manually annotated written reports were used to fine-tune a 12-layer BERT model pre-trained on a large dataset. The remaining written reports are used as test material. The extracted labels are finally used to annotate image data, training a 4-layer 3D convolutional neural network. In [29] NLP is used for adverse event detection from radiology reports and follow-up telephone call notes. In particular, hip dislocation after a primary total hip replacement [30] is used as a case study. Radiology reports are manually labeled into three categories (current dislocation, evidence of previous dislocation and no dislocation) while telephone notes are organized into two categories (evidence of previous dislocation and no dislocation). Then, the performance of different machine learning and deep learning models is compared. In [31] is observed that textual radiology reports contain relevant information for determining the likelihood of radiology signs of COVID-19 in the lungs. Machine Learning NLP approaches and SNOMED-CT reference terminology [32] are thus adopted to automatically detect COVID-19 related disorders within radiology reports.

These studies are examples of NLP applications in the medical domain. However, the typology of the texts used is remarkably different from ours: they mostly analyze medical notes, often written in a highly structured language, or abstracts, while we analyze free-text specialized manuals or papers. Finally, the purpose is different: our goal is to lay the foundations for extracting a synthetic workflow by mining descriptions of surgical procedures abundantly available in the literature, while theirs is mostly focused on helping surgeons or assistants to analyze available data.

Procedural knowledge extraction. More similar in terms of the overarching goal, but more diverse in the application domain are the studies that, similarly to our work, propose approaches for extracting procedural knowledge, however for domains other than the biomedical one. In [33], the authors tackle the problem of procedural knowledge detection in technical documentation as a classification task using Support Vector Machine with linguistic and structural features. The authors of [34] address instead the mining of cooking recipes and maintenance manuals, exploiting a CNN fed with word embeddings. Recipe for nanomaterials' synthesis has been mined in [35], where the authors use a Naïve Bayes classifier fed with features such as word counts, TF–IDF (Term Frequency–Inverse Document Frequency) and N-grams. In [36], the authors pursue the extraction of repair instructions in user-generated text from automotive web communities using linguistic and structural features fed to several machine learning methods. In [37], a Support Vector Machine is applied for extracting procedural information in technical support documentation, where procedures are typically described using lists. Also the authors of [38] address the extraction of procedural knowledge from structured instructional texts, exploiting finite-state grammars. Recently, deep-learning based NLP techniques have also been applied to extract business processes from Standard Operating Procedure documents [39].

While all these works address the extraction of procedural knowledge from written text and are thus similar to our foreseen application, they deal with typologies of textual content substantially different from the description of a surgical procedure. Troubleshooting and product documentation, cooking recipes, maintenance manuals, and repair instructions differ a lot from descriptions of surgical procedures. They are different both from the terminological point of view as well as the structural one, since these kinds of texts are structurally organized, frequently using numbered/bulleted lists, while no established standard way to describe a surgical procedure exists. In addition, surgical interventions are mainly presented in a prose-like style. Indeed, the scenario where structural features cannot be exploited is considered more challenging to tackle also in some of the presented works (c.f., e.g., [37]).

3. Method

In this section, we describe the SRL task and the neural network architecture adopted, together with the annotated resources used for supervised training. We then describe different pre-trained language models and the fine-tuning process on the downstream task of SRL. Finally, we describe the evaluation method for the proposed models. Fig. 1 summarizes the proposed approach.

3.1. Semantic role labeling

SRL is the task of labeling semantic arguments of predicates in sentences to identify *"Who"* does *"What"* to *"Whom"*, *"How"*, *"When"* and *"Where"*. Such representations are also applied for information extraction in various biomedical domains [40–42].

There are two commonly used lexical resources that make use of different typologies of semantic roles: PropBank [43] and FrameNet [44]. In this paper we use the first approach, since the literature already offers a version of PropBank specialized for the robotic-surgical domain, named RSPF (Robotic-Surgery Propositional Framebank) [5].

The typical SRL task is composed of two sub-tasks:

- Predicate identification and disambiguation: to identify each predicate in a sentence, assigning it the appropriate meaning (i.e., sense in PropBank/RSPF) in the given context, among the available ones for that predicate lemma codified in the target lexical resource;
- Argument identification and classification: to detect the argument spans or argument syntactic heads of a predicate, and to assign them the appropriate semantic role labels according to the target lexical resource.

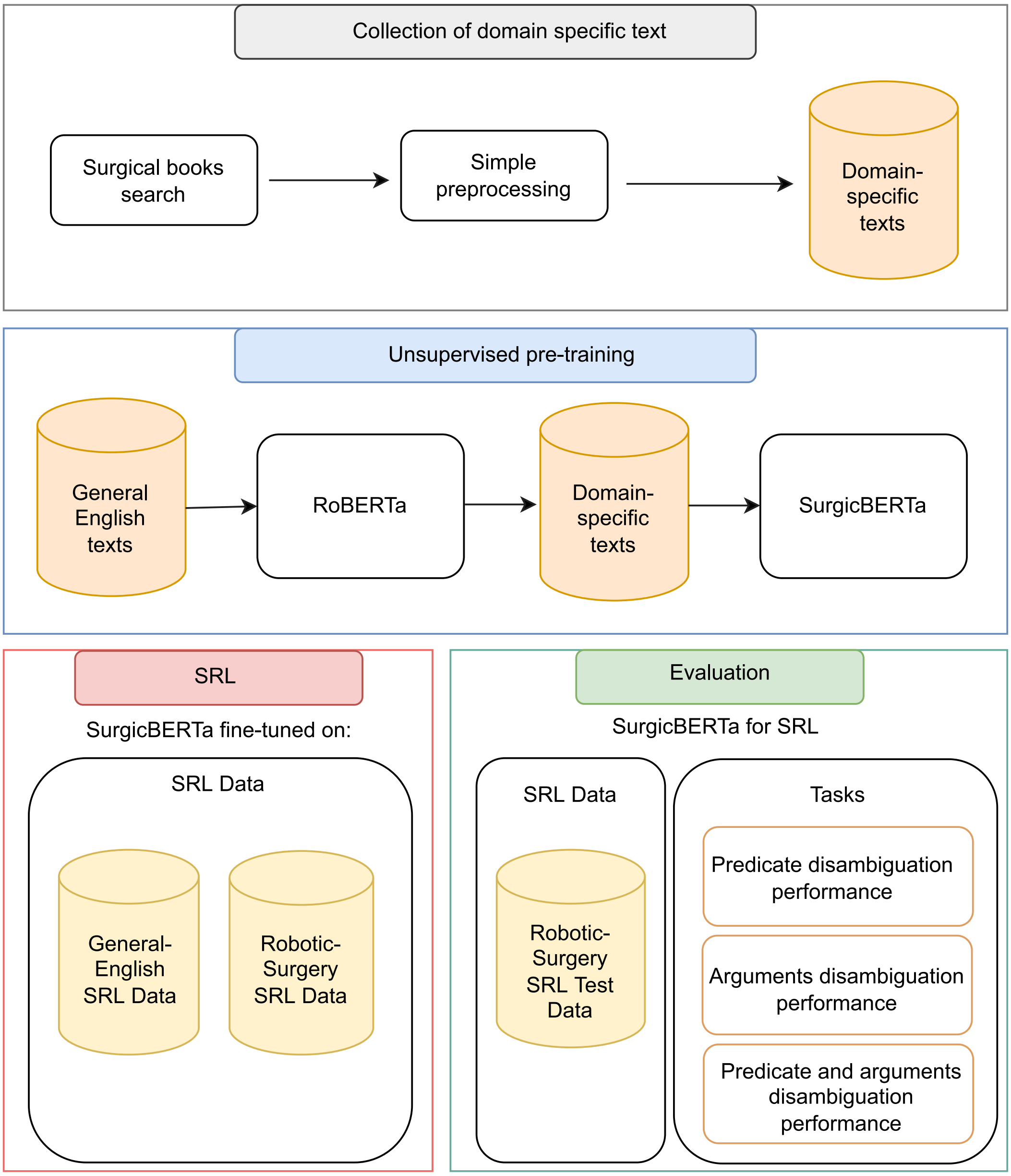
Collection of domain specific text
Surgical books search
Simple preprocessing
Domain-specific texts
Unsupervised pre-training
General English texts
RoBERTa
Domain-specific texts
SurgicBERTa
SRL
SurgicBERTa fine-tuned on:
SRL Data
General-English SRL Data
Robotic-Surgery SRL Data
Evaluation
SurgicBERTa for SRL
SRL Data
Robotic-Surgery SRL Test Data
Tasks
Predicate disambiguation performance
Arguments disambiguation performance
Predicate and arguments disambiguation performance

Fig. 1. Overview of our approach for procedural surgical knowledge extraction. The pipeline is composed of three stages (top to bottom): (i) the collection of surgical texts from the web and a simple pre-processing (gray box). These data are used to adapt RoBERTa pre-trained language model to the surgical domain. (ii) Using the data from the gray box, we adapt RoBERTa pre-trained language model to the surgical domain through unsupervised training (masked language modeling). We thus obtain a new pre-trained language model specific to the surgical domain, named SurgicBERTa (right part of the blue box); (iii) We then fine-tune SurgicBERTa in a supervised way on the downstream SRL task using general-English and surgical annotated datasets (red box); SurgicBERTa thus learns the surgical SRL task. (iv) With the performance evaluation step (green box), we evaluate the obtained model on a further test dataset consisting of SRL-style annotated surgical sentences. We evaluate the model on three different dimensions: the ability to disambiguate the sense of the predicate of the sentence, the ability to disambiguate semantic arguments and finally the ability to jointly disambiguate predicate and semantic arguments.

A common example in general English is the sentence *"Mary sold the book to John"*. In the predicate identification and disambiguation phase, SRL identifies that *"sold"* is the predicate and in this sentence it has, among the six alternative senses for *"sell"* codified in PropBank, the meaning *sell.01 - commerce: seller, giving in exchange for money*. In the argument identification and classification phase, the SRL instead produces the following output:

"[**Arg0**: Mary] [**sell.01**: sold] [**Arg1**: the book] [**Arg2**: to John]".

where, for the sense *sell.01* in PropBank, Arg0 identifies the *seller*, Arg1 the *thing sold*, and Arg2 *the buyer*.

As another example, consider the following sentence from the surgical domain, focusing on the verb *"grasp"*: *"Using the cadiere grasper (robot arm #3), grasp the soft tissues along the lesser curvature of the stomach to straighten out the lga perpendicular to the celiac axis"*. In the predicate identification and disambiguation phase, *"grasp"* is recognized as a predicate, assigning it the meaning of *grasp.02*: *"to clasp or embrace especially with the fingers or arms"*, rather than *grasp.01*: *"to take hold of, comprehend"*, in RSPF. Then, in the argument identification and classification phase, SRL produces the following output:

"[**Arg2**: Using the cadiere grasper (robot arm #3)], [**grasp.02** grasp] [**Arg1**: the soft tissues] [**Arg3**: along the lesser curvature of the stomach to straighten out the lga perpendicular to the celiac axis]".

where, for the sense *grasp.02* in RSPF, Arg2 represents the *"instrument used for grasping"*, Arg1 is the *"thing grasped"*, and Arg3 identifies an *"important spatial indication for correct grasping"*.

3.2. Annotated textual resources

Modern SRL methods rely on neural architectures and require annotated data to learn the language in a supervised way. For training, validating and testing the methods investigated in our work, we relied on two different manually annotated textual datasets for SRL: the CoNLL-2012 dataset [45], a large-scale (∼ 318k annotated SRL predicates), multi-genre general-English corpus, and a smaller dataset, specific of the robotic-surgery domain. We used the first to make the architecture learn the standard SRL task, and the second to specialize the model so that it better understands the surgical language and how to perform the SRL task in the given domain. The annotated robotic-surgery textual dataset is an extended version of the SPKS dataset [1] whose sentences have been manually annotated in a SRL PropBank style, using as framebank RSPF, which is an extension of the standard PropBank's frameset with frames describing actions and semantic roles used in the robotic-assisted surgical domain. The dataset contains sentences describing robotic-surgery procedures, thus including all traditional surgical actions plus others that are specific to robot operations. In RSPF, 24 new predicates, 22 new senses (or frames), and 63 missing arguments have been added to the standard PropBank's frameset and used to annotate SPKS sentences.[2] In total, we relied on 1559 SRL annotated sentences describing 28 surgical procedures of four different robotic-surgery domains:

- *Urology* - 51.51% of the sentences;
- *Gastrointestinal procedures* - 24.82% of the sentences;
- *Thoracic procedures* - 13.02% of the sentences;
- *Gynecology* - 10.65% of the sentences

The dataset is composed of 3601 predicates and 8601 annotated semantic arguments. In more detail, we have 460 Arg-0, 3462 Arg-1, 1079 Arg-2, 446 Arg-3, 276 Arg-4, 45 Arg-5, 9 Arg-6, 2759 ArgM of different types and 65 R-Arg related to different core arguments or arguments representing coreference.

Both datasets (CoNLL-2012 and the annotated SPKS) adhere to the PropBank way of annotating predicates and semantic arguments. In more detail, because of the difficulty of defining a universal set of semantic or thematic roles covering all types of predicates, PropBank (and thus RSPF) defines semantic roles on a verb-by-verb basis, specifying them for the different senses (i.e., frames) of a verb. Semantic arguments are numbered consecutively, starting from zero (i.e., Arg-0 Arg-1, Arg-2, Arg-3, Arg-4, Arg-5 and Arg-6). These semantic arguments are also called *"core"* because they are essential to describe the usage of the corresponding frame. In addition to the semantic roles describing the specific frame, verbs can take any of a set of general, adjunct-like arguments (ArgMs), distinguished by a given function (also called *"modifier"*). Examples of subtypes of the ArgM modifier tag are LOC (i.e., location), NEG (i.e., negation marker), PRP (i.e., purpose) and TMP (i.e., time). The complete list of ArgM modifiers is in [43].

As an example, we consider here the frame *dissect.02* that in RSPF has 6 core-roles:

- Arg-0: mad scientist, agent
- Arg-1: entity dissected
- Arg-2: instrument
- Arg-3: manner or technique
- Arg-4: spatial indications useful for the dissection
- Arg-5: until to dissect

Using this frame and the above roles, the sentence

> "The lymphatic tissue is dissected off with meticulous hemostatic and lymphatic control, using bipolar electrocautery and hem-o-lok® clips, to improve visualization".

is annotated as:

> "[**Arg-1**: The lymphatic tissue] is [**dissect.02** dissected] off [**Arg-3:** with meticulous hemostatic and lymphatic control], [**Arg-2:** using bipolar electrocautery and hem-o-lok® clips], [**ArgM-PRP**: to improve visualization.]"

3.3. The SRL neural architecture adopted

SRL is traditionally performed with data-driven methods [46]. Traditional approaches were based on classifiers trained on manually-engineered textual features: e.g., [6] proposes a statistical classifier trained using various morpho-syntactic features (e.g., governing predicate, phrase type). Recent works on SRL leverage deep neural networks, shifting from feature-engineering to architecture-engineering, with several notable approaches suggesting to perform SRL in an end-to-end fashion, relying only on raw low-level input signals (characters/tokens) fed to advanced techniques, such as multi-layer recurrent networks [47]. More recently, approaches leveraging self-attention techniques [48] and Transformer-based architectures with pre-trained language models [49] have been proposed. In this work, we follow this trend and adopt a neural approach, thus addressing the SRL task in an end-to-end fashion while testing different pre-trained language models. The pre-trained language models considered in this paper are the state-of-the-art RoBERTa [8] and BioMedRoBERTa [9], described in Section 3.5.1, and the one we contribute and release, SurgicBERTa, described in Section 3.5.2. That is, all SRL models compared in this paper share the same architecture, and differ only in the pre-trained language models used and the data on which they are pre-trained.

In more detail, the SRL models used in this paper are instantiated on top of the RoBERTa encoder (the same also used by BioMedRoBERTa and SurgicBERTa), which has been shown in much current work to achieve state-of-the-art language understanding capabilities. At its core, the system is a standard BIO tagger whose objective is to assign a label of the form `B-X` (beginning of argument with role `X`), `I-X` (continuing of argument with role `X`) or `O` (not an argument) to the tokens of the sentence, with respect to the considered predicate. Fig. 2 illustrates the neural architecture we use. First, we encode the input text using contextualized word embeddings for each token using the pre-trained language model; we then use linear transformations of the word embeddings to obtain a concatenated input for a two-layer ReLU, which is next input to a linear layer followed by softmax activation to produce a probability distribution over labels for each word (to avoid overfitting, a standard dropout layer [50] with probability 0.5 is used). To capture the sequential dependencies between labels, we use a standard CRF layer [51] to produce at testing the most probable label sequence using standard Viterbi decoding.

For training and validation, we rely on the datasets described in the previous section. The CoNLL-2012 dataset is used to train and validate the *"zero-shot"* models, while the robotic-surgery annotated dataset is used in combination with CoNLL-2012 for the *"few-shot"* and *"full fine-tuning"* models that will be described in Section 3.6. Evaluation is carried out on different test splits of the robotic-surgery annotated dataset, detailed next in Section 3.4. We remark that sentences of the test sets were never seen during the training and validation phases.

In all experiments, we inform the model about the tokens that are playing the role of predicate. Differently from [49], we do not use the gold frame sense since our purpose is also to evaluate the model's ability to correctly disambiguate the predicate meaning. The predicate disambiguation adopts a similar architecture.

[2] plus some other sentences taken from additional surgical procedures.

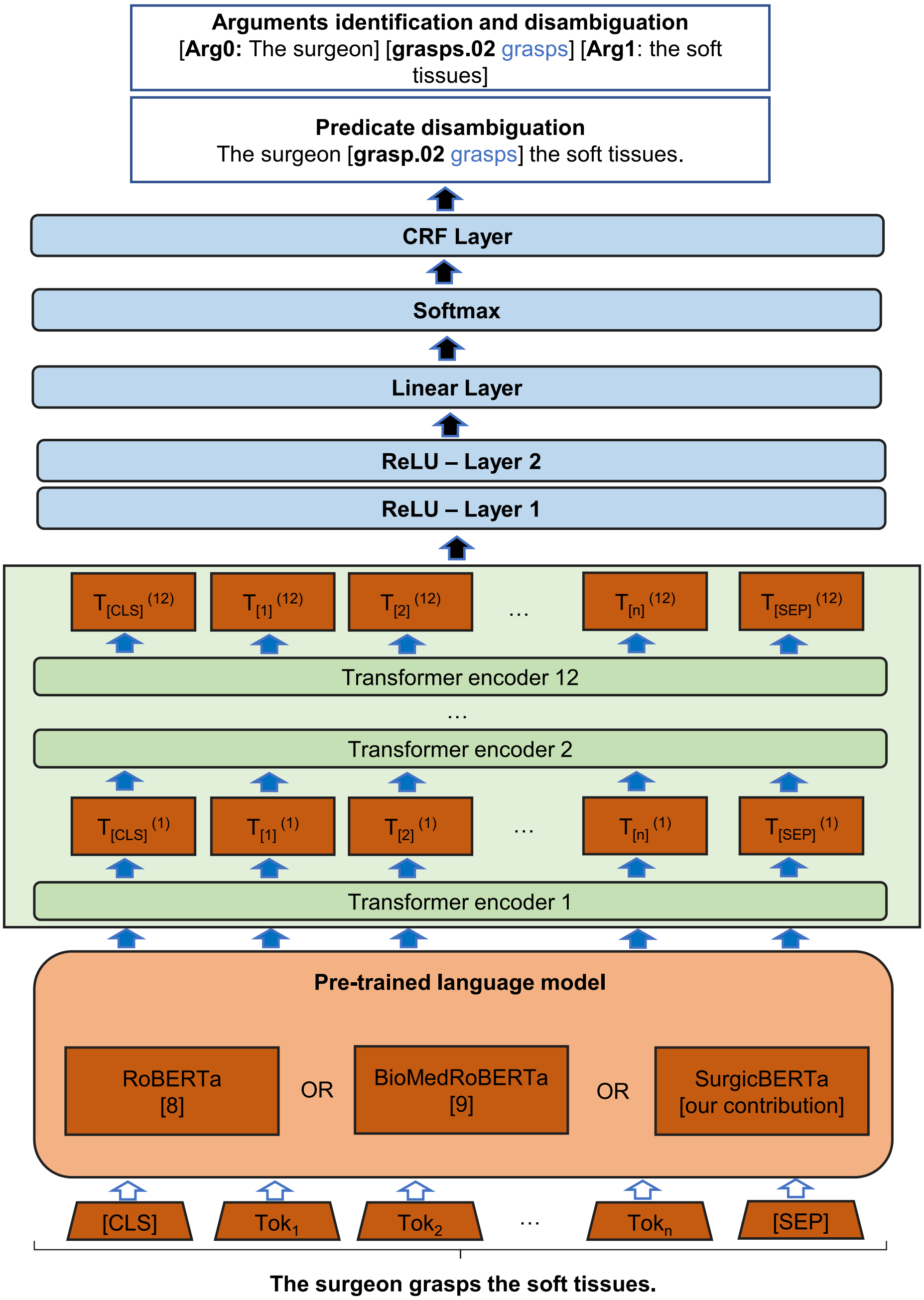

Fig. 2. The neural architecture used for SRL. Sentences are tokenized and each token is input to a pre-trained language model to produce a contextualized representation, which is then fed into ReLU layers and a linear layer. Next, a softmax layer produces a probability distribution over the labels. A CRF layer finally captures dependencies between labels by decoding the resulting representations into the most probable label sequence.

3.4. Splits of the robotic-surgery annotated dataset

Due to the high computational costs needed for training/validating/testing the SRL models, we adopted the classical evaluation protocol of manually splitting the dataset into three components (train, validation, and test) instead of following a more computationally demanding cross-validation protocol. More in detail, we split the robotic-surgery annotated dataset presented in Section 3.2 into three different combinations:

- **BAL**: the split train-test-validation is balanced between different surgical domains. The procedures are split into train-test-validation preserving the number of sentences per each domain (thoracic, gynecological, urological, gastrointestinal). Then, 80% of sentences are used to train (10% of them are removed and used to validate the dataset) and 20% as a test dataset. A similar approach is also used by [43].
- **GYN**: train and validation datasets contain all the sentences of thoracic, gastrointestinal and urological descriptions. The test dataset contains only sentences of the gynecological domain. No sentences describing gynecological surgeries were seen during the training and validation steps.
- **THO**: train and validation datasets contain all the sentences of gynecological, gastrointestinal and urological descriptions. The test dataset contains only sentences of the thoracic domain. No sentences describing thoracic surgeries were seen during the training and validation steps.

Table 1
Statistics of the different splits. The numbers outside the parenthesis represent the percentage of the corresponding semantic argument in the respective train + validation or test datasets: the sum by columns of the numbers outside the parenthesis is therefore 100. The numbers in parenthesis represent the split of the corresponding argument in the train + validation and test dataset. For each argument, the sum of the number in parenthesis of train + validation and test datasets is therefore 100. The same is for predicates (Preds in table).

Split	BAL		THO		GYN	
Pred. or arg.	Train+Val (%)	Test (%)	Train+Val (%)	Test (%)	Train+Val (%)	Test (%)
Preds	(80.20)	(19.80)	(87.46)	(12.54)	(89.28)	(10.71)
Arg-0	5.50 (81.90)	4.96 (18.10)	5.15 (81.74)	7.64 (18.26)	5.40 (87.26)	6.01 (12.74)
Arg-1	40.48 (80.21)	40.71 (19.79)	41.84 (87.22)	40.66 (12.78)	42.03 (89.31)	38.34 (10.69)
Arg-2	13.18 (83.56)	10.56 (16.44)	12.96 (87.52)	12.26 (12.48)	13.05 (89.87)	11.21 (10.13)
Arg-3	4.94 (75.95)	6.37 (24.05)	5.50 (87.26)	5.34 (12.74)	5.69 (91.93)	3.80 (8.07)
Arg-4	2.81 (69.29)	5.07 (30.71)	3.67 (90.43)	2.58 (9.57)	3.72 (93.40)	2.00 (6.60)
Arg-5	0.54 (82.22)	0.47 (17.78)	0.55 (89.13)	0.44 (10.87)	0.49 (80.43)	0.90 (19.57)
Arg-6	0.07 (55.56)	0.24 (44.44)	0.13 (0.90)	0.09 (0.10)	0.11 (0.89)	0.10 (0.11)
ArgM	32.48 (80.71)	31.62 (19.29)	30.20 (86.60)	30.99 (13.40)	29.51 (85.66)	37.64 (14.34)

The BAL split wants to investigate the ability of the method described in 3.3 to learn a general surgical domain procedural language from a limited set of annotated sentences. The GYN and THO splits[3] aim instead to verify if the annotations are general across different surgical sub-domains, i.e., if the models trained on them perform in a comparable way with the one trained with the BAL split. Table 1 summarizes some statistics about the splits.

3.5. Language models

3.5.1. State-of-the-art pre-trained models: RoBERTa and BioMedRoBERTa

We started our investigation with two different state-of-the-art pre-trained language models: one trained on general-English texts (RoBERTa) and one trained on biomedical texts (BioMedRoBERTa). RoBERTa was trained on over 160 GB of uncompressed text, with sources ranging from English language encyclopedic and news articles, to literary works and web content. Representations learned by such models generally achieve strong performance across many tasks with datasets of varying sizes drawn from a variety of sources. BioMedRoBERTa is obtained from RoBERTa via continuously pre-training on 2.68M full-text biomedical papers from S2ORC [52]. This amounts to 7.55B tokens and 47 GB of data. With this configuration we want to implicitly verify if the biomedical domain is similar to the surgical one, and if we can obtain performance improvements by simply adopting a more accurate pre-trained language model than the general-domain RoBERTa. Both pre-trained language models use the same transformer-based architecture [53]. Both are trained with a masked language modeling objective (i.e., cross-entropy loss on predicting randomly masked tokens). The word representations learned in the pre-trained models are usually reused in supervised training for a downstream task, with optional updates (fine-tuning) of the representations and network from the first stage. In our case, the downstream task is SRL.

3.5.2. A novel surgical language model: SurgicBERTa

In addition to these state-of-the-art pre-trained language models, we trained a novel model specifically for the surgical domain. Similarly to the approach followed for building BioMedRoBERTa from RoBERTa, we continuously trained RoBERTa for the masked language modeling (MLM) unsupervised task, on a large amount of surgical domain text. Under MLM, some words in a given sentence are masked and the model is expected to predict those masked words based on other words in the sentence. Such a training scheme makes this model bidirectional in nature because the representation of the masked word is learnt based on the words that occur on its left as well as right. MLM can be considered similar to autoencoding modeling which works based on constructing outcomes from unarranged or corrupted input. As the name suggests, masking works with these modeling procedures which means some words from a sequence of input or sentences are masked and the designed model has to predict the masked words to complete the sentence. For instance, given the masked sentence:

The surgeon uses a [MASK] to cut the tissue.

the models should output

The surgeon uses a [scissor] to cut the tissue.

Following successful work on domain-adaptive pre-training via language modeling [54–56], we investigate the effect of running standard MLM on domain-specific texts. We selected 300 K sentences from surgery books (7 million words) and continuously trained RoBERTa on them for MLM, obtaining SurgicBERTa, a pre-trained language model specific to the surgical language. The training sentences were selected from various books covering several heterogeneous surgical (not only robotic-surgical) sub-domains, from abdominal surgery to orthopedics to eye surgery. In line with previous work on building/adapting large language models, e.g., [9], we do not constrain the selection of the training sentences to obey any balancing of these surgical sub-domains, but simply collect as much freely accessible content as possible relevant for the surgical domain at large, since our goal is to build a model to be used ubiquitously in tasks where surgical textual content is involved (i.e., beyond the scope of the specific SRL task addressed in this paper). A very minimal pre-processing of the sentences is performed by simply removing URLs and bibliographic references.

3.6. Fine-tuning of language models on the SRL downstream task

Using the neural architecture of Section 3.3, the language models of Sections 3.5.1 and 3.5.2, the RSPF resource, the general-English CoNLL-2012 and the surgical datasets described in Section 3.2, we trained 18 different models, six for each split. In particular, for each one, we fine-tuned RoBERTa, BioMedRoBERTa and SurgicBERTa on two different scenarios:

- **Zero-Shot**: we fine-tuned the language models only on CoNLL-2012 annotated data (train and validation sets), i.e. on non-surgical data. We then evaluated obtained models on the surgical test set for the various splits;
- **Full Fine-Tuning**: starting from the fine-tuned models of the Zero-Shot scenario, we continued to fine-tune them on the train and validation sets for the different splits of the robotic-surgery annotated dataset. We then evaluated the resulting models on the corresponding surgical test set according to the split, the same used in the Zero-Shot scenario.

[3] Although any of the sub-domains in the SPKS dataset could have been chosen as a test set while training on the others, given the relatively small size of the robotic-surgery annotated dataset, we opted for testing on these two domains as they are the smaller ones and thus maximize the size of the available material (from the other domains) used for training (c.f. Section 3.2).

Transformer-based language models are known for their capability to achieve high scores also when fine-tuned with a limited amount of task-specific training material (**Few-Shot learning** [57]). This capability is particularly useful in situations of scarcity of annotated data, due to few resources or costly content annotation, such as the robotic-surgical one. We thus decided to run some experiments to assess whether this holds also for the surgical SRL task. We created various subsets of the train and validation splits for the BAL scenario of the robotic-surgical annotated dataset, having a number of sentences that are respectively of 0%, 1%, 5%, 10%, 25%, 50% and 100% of the original training and validation sets. We then trained the SurgicBERTa model on these different subsets, validating them on the same reference test set.

Following the guidelines provided by the authors of [49], we performed the fine-tuning of the models on the downstream SRL task in two stages, with the following suggested configurations:

- stage 1: fine-tuning using cross-entropy loss for 30 epochs with learning rate 3×10^{-5};
- stage 2: further fine-tuning using the combined loss for additional 5 epochs with a lower learning rate (1×10^{-5}.)

Details on the loss functions used can be found in [49].

3.7. Evaluation

Performance is evaluated according to three different dimensions:

- *argument identification and disambiguation*: the capability of assigning the correct semantic role label to the predicate arguments mentioned in the text, after identifying it. This is the traditional dimension used for bench-marking SRL tools [7,46,49], adopted also in the CoNLL-2012 Shared Task evaluation (and corresponding script);
- *predicate disambiguation*: the capability of assigning the correct RSPF frame (i.e., meaning) to the predicate in the text. In our domain setting, this evaluation is particularly useful to assess if the models are capable to discriminate the domain-specific usage of some verbs with respect to their general-English usage;
- *predicate-argument disambiguation*: the capability of assigning the correct semantic role label to the predicate arguments as well as to assign the correct sense (i.e., frame) to the corresponding predicate.

The first two dimensions correspond to the two sub-tasks described in Section 3.1, while the third one aims at combining the correctness on both dimensions. To the best of our knowledge, the assessment of this combined predicate-argument disambiguation performance was not addressed in previous works and evaluation campaigns, although we deem it particularly relevant for assessing SRL performance, especially for Propbank-style annotations: indeed, as arguments are defined in RSPF (and PropBank) according to predicate senses (i.e., different senses of the same predicate have different semantic roles), if a tool correctly predicts the role label (e.g., Arg-1) for the argument but fails to disambiguate the sense of the corresponding predicate (e.g., proposing dissect.02 instead of correct dissect.01), it basically fails in predicting the actual semantic arguments for that predicate, as it predicted a semantic role but for a different predicate sense. Note that these cases are not handled by the standard CoNLL-2012 argument disambiguation, for which the role assigned to an argument is correct independently of the disambiguated sense of the corresponding predicate.

In practice, the evaluation compares the annotations made on the sentence with the gold ones. Namely, for each token of the sentence, the predicted annotation is compared with the gold one. For the first dimension, only the labels of the arguments are considered, while in the second dimension only the labels of predicates are used. Finally, for the third dimension, the comparison is performed on enriched labels derived from the raw ones as follows: the argument label on each token (both gold and predicted) is concatenated with the label of the corresponding predicate sense so that the same annotation contains both information on the role of the argument and the predicate sense to which that role refers to. Then, for each dimension, performance is computed with standard metrics for classification tasks, i.e., precision, recall, and F1-score.

In more detail, for each classification class i (role or predicate sense label), we compute the following class metrics:

- class i true positives (TP^i): the number of arguments or predicate senses correctly predicted in class i;
- class i false positives (FP^i): the number of arguments or predicate senses predicted in class i that are in other classes according to the gold standard;
- class i false negatives (FN^i): the number of arguments or predicate senses that are in class i according to the gold standard, but are predicted in the other classes.

Starting from previous values, we then compute the following metrics:

- $Precision$ (P) $= \frac{TP^i}{TP^i+FP^i}$
- $Recall$ (R) $= \frac{TP^i}{TP^i+FN^i}$
- $F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP^i}{2*TP^i+FP^i+FN^i}$

We further compute the *Accuracy* metric (Acc), i.e., the ratio between the correctly predicted classes, divided over the test set size. For each dimension, the higher the score of the metric, the better the model performs.

3.8. Computational aspects

All models are computed using one NVIDIA RTX A6000 GPU, with 48 GB of GPU memory, with the (one-time) MLM training required for building SurgicBERTa taking approximately 8 h.

Since the compared models share the same SRL neural architecture and vary in the language model used, we observed no significant difference in the time required for fine-tuning them on the annotated dataset: indeed, each model has required approximately 20 h for this step. Although the training time is substantial, once the models have been trained, getting the annotations automatically on the test sentences is extremely fast, taking approximately 15 seconds on the largest test split, consisting of approximately 400 sentences (i.e., roughly 0.04 s per sentence). That is, exploiting the models that we publicly release, extracting surgical actions and related semantic information from a sentence is almost instantaneous.

4. Results

In this section, we report and discuss the results obtained using the methods described in Section 3. Each score reported in the section is the average over three distinct runs of the considered method.

4.1. Argument disambiguation

We first evaluate the obtained models on the traditional argument disambiguation task. The results are reported in Table 2.

The results show that having annotated domain data available is essential to improve the arguments' disambiguation performance. In fact, fine-tuning the language models with some domain data allows to significantly increase considered metrics on all splits. By focusing on the F1 metric of the BAL split, moving from a zero-shot scenario to a full fine-tuning one, we improve the performance of 0.061 for RoBERTa, of 0.065 for BioMedRoBERTa and of 0.063 for SurgicBERTa. Similar considerations hold for precision (RoBERTa +0.057; BioMedRoBERTa +0.061 and SurgicBERTa +0.054) and recall (RoBERTa +0.064; BioMedRoBERTa +0.065 and SurgicBERTa +0.072). These results confirm

Table 2
Performance (overall) on the arguments-disambiguation task for BAL, THO and GYN splits. FFT means *Full Fine-Tuning* scenario, while ZS stands for *Zero-Shot* scenario. The best scores are highlighted in bold.

Split	BAL			THO			GYN		
model	P	R	F1	P	R	F1	P	R	F1
RoBERTa$_{ZS}$	0.714	0.688	0.701	0.692	0.677	0.685	0.775	0.767	0.771
BioMedRoBERTa$_{ZS}$	0.718	**0.696**	0.707	0.708	0.684	0.696	0.788	0.777	**0.782**
SurgicBERTa$_{ZS}$	**0.724**	**0.696**	**0.710**	**0.726**	**0.700**	**0.713**	**0.827**	**0.781**	0.775
RoBERTa$_{FFT}$	0.771	0.752	0.762	0.753	0.744	0.748	0.799	0.781	0.790
BioMedRoBERTa$_{FFT}$	**0.779**	0.764	0.772	0.756	0.738	0.747	0.798	0.794	0.796
SurgicBERTa$_{FFT}$	0.778	**0.768**	**0.773**	**0.759**	**0.749**	**0.753**	**0.813**	**0.796**	**0.804**

that using domain annotated data helps the models to both improve the proportion of positive identifications that was actually correct and the proportion of actual positives that was identified correctly. This is in line with what was expected: being RSPF an extension of PropBank for the surgical domain, the CoNLL-2012 dataset, the only SRL training material used for the zero-shot models, does not contain annotated examples for some of the labels of RSPF (the ones in RSPF but not in PropBank), and thus it will not be able to predict them on the test set, where some of these labels are likely to occur. Furthermore, the domain annotated data is fundamental to accurately understand the surgical procedural language which often has different needs than those of general-English [5].

Similar considerations apply also for the performance on the THO split: the full fine-tuning improves precision (RoBERTa +0.061; BioMedRoBERTa +0.048 and SurgicBERTa +0.033), recall (RoBERTa +0.067; BioMedRoBERTa +0.054 and SurgicBERTa +0.049) and F1-score (RoBERTa +0.063; BioMedRoBERTa +0.051 and SurgicBERTa +0.040) for all considered models. The improvement between zero-shot and full-fine tuning is comparable to the one observed for the BAL split. Full fine-tuning typically improves the performance over zero-shot learning also on the GYN split, although the improvement is somehow restrained with respect to the other two splits: precision (RoBERTa +0.024; BioMedRoBERTa +0.010 and SurgicBERTa −0.014), recall (RoBERTa +0.014; BioMedRoBERTa +0.017 and SurgicBERTa +0.016) and F1-score (RoBERTa +0.019; BioMedRoBERTa +0.014 and SurgicBERTa +0.029). This minor improvement may be due to the presence of fewer sentences in the GYN split that require annotation using the RSPF specializations (i.e., those labels in RSPF but not in PropBank): this is somehow confirmed by the significantly higher values obtained with zero-shot on GYN than on the other two splits. We can thus answer RQ1 and RQ2: injecting domain sentences in the training step helps to substantially improve performance in all compared scenarios (RQ2), also when leveraging general-English and biomedical models (RQ1), whose zero-shot scores are clearly lower than the full fine-tuned ones. Also RQ5 has a positive answer since the improvement from zero-shot to full fine-tuning is comparable between the different splits, showing that the models perform reasonably well also when tested on surgical sub-domains not seen during the training.

Note that SurgicBERTa achieves the best results in both the zero-shot and full fine-tuning scenarios for almost all metrics of all splits.[4] This confirms that using unsupervised domain adaptation techniques such as MLM can improve performance even in the presence of few or no annotated data. It is interesting to note that SurgicBERTa also improves performance compared to BioMedRoBERTa which has been specialized on biomedical domain texts. This means that the procedural robotic-surgical domain, which is a specialized subset of the biomedical one, uses a "distinct" language that deserves appropriate, specialized training resources to be adequately covered by language models. We can thus positively answer RQ4.

Table 3 goes deeper in the analysis and compares the fine-grained performance, argument-by-argument, by the baseline model (i.e., RoBERTa in the zero-shot scenario - RoBERTa$_{ZS}$) with those obtained by the best model for the BAL split (i.e., SurgicBERTa in a full fine-tuning scenario - SurgicBERTa$_{FFT}$). The detailed results show that full-fine tuning for the BAL split, improves the disambiguation of almost all core and modifier arguments. The most substantial improvements are among the core numbered arguments i.e., Arg-N with $N \in 0..6$. Quite often, especially for $N \geq 3$, these are the ones not present in the standard PropBank but introduced in RSPF, and therefore are very specialized arguments of the surgical domain never seen in CoNLL-2012 data. This, again, answers RQ1, since although the zero-shot scenario with RoBERTa obtains acceptable results, using more specific language models and annotated data allows for improved performance.

4.2. Predicate and predicate-argument disambiguation

Table 4 shows the results of the other two dimensions considered in our analysis, i.e., predicate disambiguation and predicate-argument disambiguation. For predicate disambiguation, as the used SRL tool was configured to work with gold predicate mentions (i.e., having an oracle that predicts whether a token denotes a predicate or not),[5] for predicate disambiguation we only report the accuracy score, as in this setting, by definition, P=R=F1=Acc.

Similar considerations as the one reported for argument disambiguation hold also for these two assessments: using domain annotations allows to improve the performance of the models. The improvements are comparable and very noticeable for the BAL and THO splits, while they are less substantial in the GYN split. Also for the predicate disambiguation and for the predicate-argument disambiguation tasks, using a domain language model (i.e., SurgicBERTa) often improves performance. The most substantial improvements are achieved within the full-fine tuning scenario. Again, this confirms the trends of the data observed on argument disambiguation, thus confirming the answers for RQ1, RQ2, RQ4, and RQ5.

Furthermore, note that the scores for argument disambiguation in Table 2 are substantially lower than the ones for predicate-argument disambiguation reported in Table 4. For example, SurgicBERTa$_{FFT}$ obtains an F1 of 0.773 for argument disambiguation in BAL split, and only a 0.732 (i.e., −0.041) in predicate-arguments disambiguation. The difference in the scores between argument disambiguation and predicate-arguments disambiguation is even larger in the Zero-Shot scenario e.g., 0.701 vs 0.534 for RoBERTa$_{ZS}$. That is, in many cases, while the argument label proposed by the models may be correct per se (i.e., ignoring the predicate to which the argument refers), it actually denotes the argument label for a wrong predicate sense, and therefore a

[4] The only exception is in the zero-shot scenario for the F1 metric of the GYN split, where BioMedRoBERTa attains a slightly better score.

[5] Note that this is by no means a limitation of the comparison conducted in our work as: (i) predicates can be easily spotted via part-pf-speech tagging, considering only the tokens labeled as Verb, or Proper Nouns having specific suffixes (e.g., -ize, -ation); and (ii), this applies for all the models considered in the assessment.

Table 3
A fine-grained comparison between a baseline model and the best model for the argument disambiguation task. Best F1 scores are highlighted in bold.

Model	RoBERTa$_{ZS}$			SurgicBERTa$_{FFT}$		
Argument	P	R	F1	P	R	F1
Arg-0	0.696	0.655	0.675	0.879	0.691	**0.773**
Arg-1	0.903	0.890	0.896	0.911	0.926	**0.919**
Arg-2	0.647	0.553	0.596	0.671	0.603	**0.635**
Arg-3	0.000	0.000	0.000	0.554	0.380	**0.451**
Arg-4	0.000	0.000	0.000	0.614	0.628	**0.621**
Arg-5	0.000	0.000	0.000	0.000	0.000	0.000
Arg-6	0.000	0.000	0.000	1.000	0.250	**0.400**
ArgM-ADJ	0.000	0.000	0.000	0.000	0.000	0.000
ArgM-ADV	0.564	0.585	**0.574**	0.553	0.491	0.520
ArgM-CAU	0.500	1.000	**0.667**	0.500	1.000	**0.667**
ArgM-DIR	0.154	0.240	0.188	0.292	0.280	**0.286**
ArgM-DIS	0.500	0.286	0.364	0.429	0.429	**0.429**
ArgM-EXT	0.500	1.000	**0.667**	0.500	1.000	**0.667**
ArgM-GOL	0.000	0.000	0.000	0.000	0.000	0.000
ArgM-LOC	0.381	0.500	0.432	0.436	0.578	**0.514**
ArgM-MNR	0.267	0.722	0.390	0.544	0.681	**0.605**
ArgM-MOD	0.988	0.976	0.982	0.988	1.000	**0.994**
ArgM-NEG	1.000	1.000	**1.000**	1.000	1.000	**1.000**
ArgM-PNC	0.000	0.000	0.000	0.000	0.000	0.000
ArgM-PRD	0.000	0.000	0.000	0.000	0.000	0.000
ArgM-PRP	0.754	0.754	**0.754**	0.708	0.807	**0.754**
ArgM-TMP	0.827	0.865	0.845	0.865	0.865	**0.865**
R-Arg0	1.000	1.000	**1.000**	1.000	1.000	**1.000**
R-Arg1	0.857	1.000	**0.923**	0.857	1.000	**0.923**
R-Arg2	1.000	1.000	**1.000**	0.000	0.000	0.741
R-ArgM-LOC	1.000	1.000	**1.000**	0.500	1.000	0.667

Table 4
Performance (overall) on the predicate disambiguation and predicate-argument disambiguation tasks for BAL, THO and GYN splits. The best scores are highlighted in bold.

SPLIT	BAL				THO				GYN			
TASK	Predicate	Predicate-Arguments			Predicate	Predicate-Arguments			Predicate	Predicate-Arguments		
MODEL	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
RoBERTa$_{ZS}$	0.731	0.544	0.525	0.534	0.769	0.555	0.543	0.549	**0.835**	**0.649**	0.642	**0.645**
BioMedRoBERTa$_{ZS}$	**0.748**	0.560	0.543	0.551	**0.777**	**0.573**	**0.555**	**0.564**	0.810	0.641	0.632	0.636
SurgicBERTa$_{ZS}$	0.735	**0.565**	**0.544**	**0.555**	0.732	0.559	0.540	0.549	0.827	0.646	**0.643**	**0.645**
RoBERTa$_{FFT}$	0.907	0.706	0.689	0.697	**0.910**	0.680	0.672	0.676	0.930	0.745	0.729	0.737
BioMedRoBERTa$_{FFT}$	0.897	0.707	0.694	0.700	0.887	0.669	0.653	0.661	0.935	0.752	**0.748**	**0.750**
SurgicBERTa$_{FFT}$	**0.925**	**0.737**	**0.727**	**0.732**	**0.910**	**0.690**	**0.680**	**0.685**	**0.938**	**0.756**	0.741	0.749

wrong argument label in the end, since argument labels are predicate-sense specific in resources such as PropBank. This further confirms the relevance of considering the proposed joint predicate-argument disambiguation performance in SRL evaluations, in addition to the standard (and independent) argument disambiguation and predicate disambiguation.

4.3. Few-shot learning

Finally, Fig. 3 shows the few-shot learning curve of the SurgicBERTa model, obtained by varying the number of training (and validation) sentences. This assessment allows us to address RQ3.

The curve shows that if the number of added domain annotations is too small, a detrimental effect is obtained for all the analyzed metrics (P, R, and F1). However, with at least 15% of the training material (approximately 190 sentences), the performance constantly grows as annotations are added. Indeed, the curve shows a positive trend also when using all the available domain annotated material (i.e., full fine-tuning), thus suggesting that further improvements are likely by injecting additional annotated examples. However, we remark that annotating data for the SRL task in the surgical domain is quite demanding, requiring both linguistic and surgical skills, and its cost is not negligible. This analysis answers RQ3.

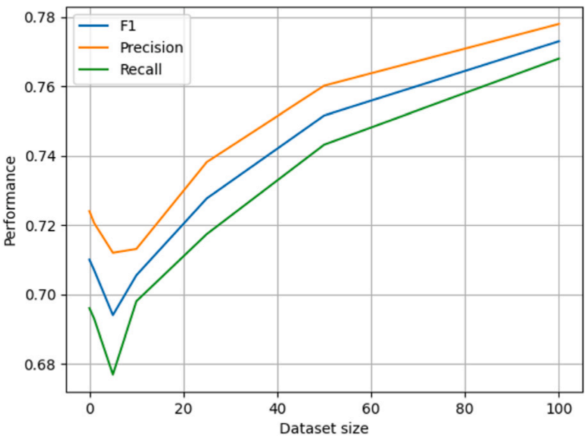

Fig. 3. Few-Shot performance of SurgicBERTa model by varying the number of training (and validation) domain sentences.

5. Conclusions

In this paper, we tackled the problem of automatically extracting procedural surgical knowledge from available surgical text materials, such as textbooks and academic papers. Given a text, the goal is to extract structured information on the surgical actions described, the agents performing them, the anatomical parts involved, the tools used, and so on. We proposed to frame the problem as a SRL task and to apply a state-of-the-art approach based on Transformer-based language models. In detail, we experimented with different models: RoBERTa (general-English), BioMedRoBERTa (biomedical domain), and the newly contributed SurgicBERTa, a pre-trained language model that we specifically developed exploiting a large collection of textual content from the surgical domain. We assessed the performance of the models in different, classical scenarios: the zero-shot scenario, where no domain-specific SRL training data is used, and the full fine-tuning scenario, where the models are additionally trained with SRL annotated sentences according to the predicate and roles defined in RSPF, a recently proposed PropBank-style resource covering the typical actions (and related information) of the surgical domain.

Results show that: (i) existing state-of-the-art tools, trained on general-English data, have low performance in extracting structured procedural content in robotic-surgery procedural texts; (ii) exploiting language models unsupervisely trained on domain-related (BioMedRoBERTa) or domain-specific data (SurgicBERTa) helps to improve the SRL performance even in the zero-shot scenario; (iii) supervised training with domain-specific SRL data substantially improves the performance of all models on all the SRL evaluation dimensions investigated, i.e., predicate disambiguation, argument disambiguation, and predicate-argument disambiguation. This seems to suggest that for adapting these general SRL methods to unexplored, specific domains like the surgical one, some domain-specific SRL manual annotation effort should likely be considered to extract high quality structured procedural information, i.e., predicates and related arguments, even if such manual annotation activity is inevitably costly, especially in highly specialized domains, like the surgical one, as both domain-specific and linguistics skills are needed to carry out the labeling.

To the best of our knowledge, this is the first work experimenting with information extraction algorithms to tackle the extraction of structured procedural knowledge in the surgical domain. As a future work we will expand the annotated dataset with annotations from additional surgical sub-domains than the ones considered so far. The specialized models will be used to develop methods to summarize and simplify surgical texts, and to inform autonomous robotic systems or robot-assisted surgery with information extracted from textbooks. We will also investigate and experiment with techniques to reduce the manual annotation costs for SRL, such as active learning. Finally, our ultimate, challenging goal is to tackle the extraction of the full workflow of a surgical intervention, that is, organizing the surgical actions extracted from the text of the procedure according to the causal/temporal relations expressed in it, so that an autonomous agent can directly replicate them exactly as they are intended to be performed in a successful surgical intervention.

Downloads

The SurgicBERTa$_{FFT}$ language model together with the code and resources used in our experiments are available under an open license at https://gitlab.com/altairLab/surgical_srl

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 742671 "ARS").

References

[1] M. Bombieri, M. Rospocher, D. Dall'Alba, P. Fiorini, Automatic detection of procedural knowledge in robotic-assisted surgical texts, Int. J. Comput. Assist. Radiol. Surg. 16 (8) (2021) 1287–1295, http://dx.doi.org/10.1007/s11548-021-02370-9.

[2] D. Meli, M. Sridharan, P. Fiorini, Inductive learning of answer set programs for autonomous surgical task planning, Mach. Learn. 110 (7) (2021) 1739–1763, http://dx.doi.org/10.1007/s10994-021-06013-7.

[3] S. Ramesh, D. Dall'Alba, C. Gonzalez, T. Yu, P. Mascagni, D. Mutter, J. Marescaux, P. Fiorini, N. Padoy, Multi-task temporal convolutional networks for joint recognition of surgical phases and steps in gastric bypass procedures, Int. J. Comput. Assist. Radiol. Surg. 16 (7) (2021) 1111–1119, http://dx.doi.org/10.1007/s11548-021-02388.

[4] B. Gibaud, G. Forestier, C. Feldmann, G. Ferrigno, P. Gonçalves, T. Haidegger, C. Julliard, D. Katić, H. Kenngott, L. Maier-Hein, K. März, E. De Momi, D. Nagy, H. Nakawala, J. Neumann, T. Neumuth, J. Balderrama, S. Speidel, M. Wagner, P. Jannin, Toward a standard ontology of surgical process models, Int. J. Comput. Assist. Radiol. Surg..

[5] M. Bombieri, M. Rospocher, S.P. Ponzetto, P. Fiorini, The Robotic Surgery Procedural Framebank, in: Proceedings of the Thirteenth International Conference on Language Resources and Evaluation, LREC 2022, European Language Resources Association (ELRA, Marseille, France, 2022, pp. 3950–3959.

[6] D. Gildea, D. Jurafsky, Automatic labeling of semantic roles, Comput. Linguist. 28 (3) (2002) 245–288, http://dx.doi.org/10.1162/089120102760275983.

[7] X. Carreras, L. Màrquez, Introduction to the conll-2005 shared task: Semantic role labeling, in: I. Dagan, D. Gildea (Eds.), Proceedings of the Ninth Conference on Computational Natural Language Learning, CoNLL 2005, Ann Arbor, Michigan, USA, June 29-30 2005, ACL, 2005, pp. 152–164.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692.

[9] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N.A. Smith, Don't stop pretraining: Adapt language models to domains and tasks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 8342–8360, http://dx.doi.org/10.18653/v1/2020.acl-main.740, Online.

[10] K. Bretonnel Cohen, D. Demner-Fushman, Biomedical Natural Language Processing, John Benjamins, 2014.

[11] D. Demner-Fushman, K.B. Cohen, S. Ananiadou, J. Tsujii (Eds.), Proceedings of the 21st Workshop on Biomedical Language Processing, Association for Computational Linguistics, Dublin, Ireland, 2022.

[12] S. Wu, K. Roberts, S. Datta, J. Du, Z. Ji, Y. Si, S. Soni, Q. Wang, Q. Wei, Y. Xiang, B. Zhao, H. Xu, Deep learning in clinical natural language processing: A methodical review, J. Am. Med. Inform. Assoc. 27 (3) (2020) 457–470, http://dx.doi.org/10.1093/jamia/ocz200.

[13] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Y. LeCun Y. Bengio (Ed.), 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4 2013, Workshop Track Proceedings, 2013, pp. 1–12.

[14] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162.

[15] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146, http://dx.doi.org/10.1162/tacl_a_00051.

[16] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, M. McDermott, Publicly available clinical BERT embeddings, in: Proceedings of the 2nd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 72–78.

[17] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, Biobert: A pre-trained biomedical language representation model for biomedical text mining, Bioinform. 36 (4) (2020) 1234–1240, http://dx.doi.org/10.1093/bioinformatics/btz682.

[18] A.E. Johnson, T.J. Pollard, L. Shen, L.-W.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R.G. Mark, Mimic-iii, a freely accessible critical care database, Sci. Data 3, http://dx.doi.org/10.1038/sdata.2016.35.

[19] Ö. Uzuner, B.R. South, S. Shen, S.L. DuVall, 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text, J. Am. Med. Inform. Assoc. 18 (5) (2011) 552–556, http://dx.doi.org/10.1136/amiajnl-2011-000203.

[20] S. Bethard, L. Derczynski, G. Savova, J. Pustejovsky, M. Verhagen, SemEval-2015 task 6: Clinical TempEval, in: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval 2015, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 806–814, http://dx.doi.org/10.18653/v1/S15-2136.

[21] S. Kulshrestha, D. Dligach, C. Joyce, M.S. Baker, R. Gonzalez, A.P. O'Rourke, J.M. Glazer, A. Stey, J.M. Kruser, M.M. Churpek, M. Afshar, Prediction of severe chest injury using natural language processing from the electronic health record, Injury 52 (2) (2021) 205–212, http://dx.doi.org/10.1016/j.injury.2020.10.094.

[22] E. Sagheb, T. Ramazanian, A.P. Tafti, S. Fu, W.K. Kremers, D.J. Berry, D.G. Lewallen, S. Sohn, H. Maradit Kremers, Use of natural language processing algorithms to identify common data elements in operative notes for knee arthroplasty, J. Arthroplasty 36 (3) (2021) 922–926, http://dx.doi.org/10.1016/j.arth.2020.09.029.

[23] S. Fu, C.C. Wyles, D.R. Osmon, M.L. Carvour, E. Sagheb, T. Ramazanian, W.K. Kremers, D.G. Lewallen, D.J. Berry, S. Sohn, H.M. Kremers, Automated detection of periprosthetic joint infections and data elements using natural language processing, J. Arthroplasty 36 (2) (2021) 688–692, http://dx.doi.org/10.1016/j.arth.2020.07.076.

[24] A.V. Karhade, M.E. Bongers, O.Q. Groot, E.R. Kazarian, T.D. Cha, H.A. Fogel, S.H. Hershman, D.G. Tobert, A.J. Schoenfeld, C.M. Bono, J.D. Kang, M.B. Harris, J.H. Schwab, Natural language processing for automated detection of incidental durotomy, Spine J. 20 (5) (2020) 695–700, http://dx.doi.org/10.1016/j.spinee.2019.12.006.

[25] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[26] S.-k. Song, H.-s. Oh, S.H. Myaeng, S.-p. Choi, H.-w. Chun, Y.-s. Choi, C.-h. Jeong, Procedural knowledge extraction on medline abstracts, in: N. Zhong, V. Callaghan, A.A. Ghorbani, B. Hu (Eds.), Active Media Technology, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 345–354.

[27] E. Hagberg, D. Hagerman, R. Johansson, N. Hosseini, J. Liu, E. Björnsson, J. Alvén, O. Hjelmgren, Semi-supervised learning with natural language processing for right ventricle classification in echocardiography—A scalable approach, Comput. Biol. Med. 143 (2022) 105282, http://dx.doi.org/10.1016/j.compbiomed.2022.105282.

[28] L.A. Zornoff, H. Skali, M.A. Pfeffer, M. St. John Sutton, J.L. Rouleau, G.A. Lamas, T. Plappert, J.R. Rouleau, L.A. Moyé, S.J. Lewis, E. Braunwald, S.D. Solomon, Right ventricular dysfunction and risk of heart failure and mortality after myocardial infarction, J. Am. Coll. Cardiol. 39 (9) (2002) 1450–1455, http://dx.doi.org/10.1016/S0735-1097(02)01804-1.

[29] A. Borjali, M. Magnéli, D. Shin, H. Malchau, O.K. Muratoglu, K.M. Varadarajan, Natural language processing with deep learning for medical adverse event detection from free-text medical narratives: A case study of detecting total hip replacement dislocation, Comput. Biol. Med. 129 (2021) 104140, http://dx.doi.org/10.1016/j.compbiomed.2020.104140.

[30] J. Parvizi, E. Picinic, P.F. Sharkey, Revision total hip arthroplasty for instability: Surgical techniques and principles, Instr. Course Lect. 58 (2009) 183–191.

[31] P. López-Úbeda, M.C. Díaz-Galiano, T. Martín-Noguerol, A. Luna, L.A. Ureña-López, M.T. Martín-Valdivia, Covid-19 detection in radiological text reports integrating entity recognition, Comput. Biol. Med. 127, http://dx.doi.org/10.1016/j.compbiomed.2020.104066, cited by: 21; All Open Access, Bronze Open Access, Green Open Access.

[32] K.A. Spackman, K.E. Campbell, R.A. Côté, SNOMED RT: A reference terminology for health care, in: AMIA 1997, American Medical Informatics Association Annual Symposium, , Nashville, TN, USA, October 25-29 1997, AMIA, 1997, pp. 640–644.

[33] S. Agarwal, S. Atreja, V. Agarwal, Extracting procedural knowledge from technical documents, arXiv preprint arXiv:2010.10156, arXiv:2010.10156.

[34] C. Qian, L. Wen, A. Kumar, L. Lin, L. Lin, Z. Zong, S. Li, J. Wang, An approach for process model extraction by multi-grained text classification, in: S. Dustdar, E. Yu, C. Salinesi, D. Rieu, V. Pant (Eds.), Advanced Information Systems Engineering, Springer International Publishing, Cham, 2020, pp. 268–282.

[35] H. Yang, C.A. Aguirre, M.F. De La Torre, D. Christensen, L. Bobadilla, E. Davich, J. Roth, L. Luo, Y. Theis, A. Lam, T.Y. Han, D. Buttler, W.H. Hsu, Pipelines for procedural information extraction from scientific literature: Towards recipes using machine learning and data science, in: 2019 International Conference on Document Analysis and Recognition Workshops, Vol. 2, ICDARW, 2019, pp. 41–46.

[36] T. Wambsganß, H. Fromm, Mining user-generated repair instructions from automotive web communities, in: B. T.X. (Ed.), Proceedings of the Annual Hawaii International Conference on System Sciences, Vol. 2019-January, IEEE Computer Society, 2019, pp. 1184–1193.

[37] A. Gupta, A. Khosla, G. Singh, G. Dasgupta, Mining procedures from technical support documents, arXiv:1805.09780 arXiv:1805.09780.

[38] Z. Zhang, P. Webster, V. Uren, A. Varga, F. Ciravegna, Automatically extracting procedural knowledge from instructional texts using natural language processing, in: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC'12, European Language Resources Association (ELRA), Istanbul, Turkey, 2012, pp. 520–527.

[39] P. Bellan, M. Dragoni, C. Ghidini, Extracting business process entities and relations from text using pre-trained language models and in-context learning, in: J.P.A. Almeida, D. Karastoyanova, G. Guizzardi, M. Montali, F.M. Maggi, C.M. Fonseca (Eds.), Enterprise Design, Operations, and Computing - 26th International Conference, EDOC 2022, Bozen-Bolzano, Italy, October 3-7 2022, Proceedings, in: Lecture Notes in Computer Science, vol. 13585, Springer, 2022, pp. 182–199, http://dx.doi.org/10.1007/978-3-031-17604-3_11.

[40] T. Barnickel, J. Weston, R. Collobert, H.-W. Mewes, V. Stümpflen, Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts, PLoS One 4 (7) (2009) 1–6.

[41] S. Bethard, Z. Lu, J.H. Martin, L. Hunter, Semantic role labeling for protein transport predicates, BMC Bioinformatics 9 (1) (2008) 1–15.

[42] F. Eckert, M. Neves, Semantic role labeling tools for biomedical question answering: A study of selected tools on the bioASQ datasets, in: Proceedings of the 6th BioASQ Workshop A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 11–21, http://dx.doi.org/10.18653/v1/W18-5302.

[43] M. Palmer, P.R. Kingsbury, D. Gildea, The proposition bank: An annotated corpus of semantic roles, Comput. Linguist. 31 (1) (2005) 71–106, http://dx.doi.org/10.1162/0891201053630264.

[44] C.J. Fillmore, C.F. Baker, A frames approach to semantic analysis, in: The Oxford Handbook of Linguistic Analysis, 2009, pp. 313–340.

[45] S. Pradhan, A. Moschitti, N. Xue, H.T. Ng, A. Björkelund, O. Uryupina, Y. Zhang, Z. Zhong, Towards robust linguistic analysis using ontonotes, in: J. Hockenmaier, S. Riedel (Eds.), Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9 2013, ACL, 2013, pp. 143–152.

[46] L. Màrquez, X. Carreras, K.C. Litkowski, S. Stevenson, Semantic role labeling: An introduction to the special issue, Comput. Linguist. 34 (2) (2008) 145–159, http://dx.doi.org/10.1162/coli.2008.34.2.145.

[47] L. He, K. Lee, M. Lewis, L. Zettlemoyer, Deep semantic role labeling: What works and what's next, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 473–483, http://dx.doi.org/10.18653/v1/P17-1044.

[48] E. Strubell, P. Verga, D. Andor, D. Weiss, A. McCallum, Linguistically-informed self-attention for semantic role labeling, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, Association for Computational Linguistics, 2018, pp. 5027–5038.

[49] T. Li, P.A. Jawale, M. Palmer, V. Srikumar, Structured tuning for semantic role labeling, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8402–8412.

[50] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (56) (2014) 1929–1958.

[51] N. Reimers, I. Gurevych, Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 338–348, http://dx.doi.org/10.18653/v1/D17-1035.

[52] K. Lo, L.L. Wang, M. Neumann, R. Kinney, D.S. Weld, S2ORC: The semantic scholar open research corpus, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4969–4983.

[53] X. Zhang, H. Yang, E.F.Y. Young, Attentional transfer is all you need: Technology-aware layout pattern generation, in: 58th ACM/IEEE Design Automation Conference, DAC 2021, San Francisco, CA, USA, December 5-9 2021, IEEE, 2021, pp. 169–174.

[54] O.J. Bear Don't Walk IV, T. Sun, A. Perotte, N. Elhadad, Clinically relevant pretraining is all you need, J. Am. Med. Inform. Assoc. 28 (9) (2021) 1970–1976.

[55] S. Zhou, N. Wang, L. Wang, H. Liu, R. Zhang, CancerBERT: A cancer domain-specific language model for extracting breast cancer phenotypes from electronic health records, J. Am. Med. Inform. Assoc..

[56] K. Xie, R.S. Gallagher, E.C. Conrad, C.O. Garrick, S.N. Baldassano, J.M. Bernabei, P.D. Galer, N.J. Ghosn, A.S. Greenblatt, T. Jennings, A. Kornspun, C.V. Kulick-Soper, J.M. Panchal, A.R. Pattnaik, B.H. Scheid, D. Wei, M. Weitzman, R. Muthukrishnan, J. Kim, B. Litt, C.A. Ellis, D. Roth, Extracting seizure frequency from epilepsy clinic notes: A machine reading approach to natural language processing, J. Am. Med. Inform. Assoc. 29 (5) (2022) 873–881.

[57] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December (2020) 6-12, Virtual, 2020, pp. 1877–1901.