Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

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

FlauBERT vs. CamemBERT: Understanding patient's answers by a French medical chatbot

Corentin Blanc [a,b,c,d,e,*], Alexandre Bailly [a,b,c,d,e], Élie Francis [a], Thierry Guillotin [a], Fadi Jamal [f], Béchara Wakim [g], Pascal Roy [b,c,d,e]

[a] Everteam Software, Lyon, France
[b] Université de Lyon, Lyon, France
[c] Université Lyon 1, Villeurbanne, France
[d] Service de Biostatistique-Bioinformatique, Pôle Santé Publique, Hospices Civils de Lyon, Lyon, France
[e] Équipe Biostatistique-Santé, Laboratoire de Biométrie et Biologie Évolutive, CNRS UMR 5558, Villeurbanne, France
[f] IzyCardio, Lyon, France
[g] Mediapps Innovation, Lyon, France

ARTICLE INFO

Keywords:
Intent and slot prediction
FlauBERT
CamemBERT
Language models
Natural Language Understanding
Neural network architectures

ABSTRACT

In a number of circumstances, obtaining health-related information from a patient is time-consuming, whereas a chatbot interacting efficiently with that patient might help saving health care professional time and better assisting the patient. Making a chatbot understand patients' answers uses Natural Language Understanding (NLU) technology that relies on 'intent' and 'slot' predictions. Over the last few years, language models (such as BERT) pre-trained on huge amounts of data achieved state-of-the-art intent and slot predictions by connecting a neural network architecture (e.g., linear, recurrent, long short-term memory, or bidirectional long short-term memory) and fine-tuning all language model and neural network parameters end-to-end. Currently, two language models are specialized in French language: FlauBERT and CamemBERT. This study was designed to find out which combination of language model and neural network architecture was the best for intent and slot prediction by a chatbot from a French corpus of clinical cases. The comparisons showed that FlauBERT performed better than CamemBERT whatever the network architecture used and that complex architectures did not significantly improve performance vs. simple ones whatever the language model. Thus, in the medical field, the results support recommending FlauBERT with a simple linear network architecture.

1. Introduction

During a first patient's consultation or session of specialized care, the interview with a health care professional has to collect a non-negligible amount of data that are essential to establish a diagnosis and initiate or update a plan of care. A part of this essential task may be nevertheless time-consuming. A chatbot (i.e., a conversational interface able to interact with humans) might then help saving time and speeding patient management.

A medical chatbot interacts with a patient via natural language processing (NLP); i.e., an artificial intelligence technology that allows computers to process human speech. Such a chatbot is able to collect data from a patient's speech using an auto-guided dialogue. That chatbot's function involves three iterative tasks: understanding a patient's answer, choosing the next information to obtain according to the previous one, and generating the corresponding question. The present work focuses on the first task; i.e., the Natural Language Understanding (NLU).

NLU is a sub-field of NLP that deals only with understanding human natural language through building a formal representation of the meaning of speech. One possible formal representation aims to organize the information present in a simple sentence by splitting it into a single 'intent' and several 'slots'. The intent relates to the type of information (e.g., socio-demographic characteristic, symptom, etc.), whereas a slot is an accurate datum that one or more words add to enrich the intent of the sentence. For example, age is a slot that contributes to a socio-demographic intent and the date of symptom onset is a slot that contributes to the symptom intent. In such a formal representation, intent

[*] Corresponding author at: Everteam Software, 17 quai Joseph Gillet, F-69004 Lyon, France.
E-mail address: c.blanc@everteam.com (C. Blanc).

https://doi.org/10.1016/j.artmed.2022.102264
Received 10 June 2021; Received in revised form 15 February 2022; Accepted 23 February 2022
Available online 2 March 2022

and slot prediction consists in spotting in a simple sentence a single intent and one or more related slots. A very simple example is given in Table 1.

Over the past few years, language models pre-trained on huge amounts of data emerged. These models are able to encode each word by focusing on its context within a sentence. The well-known BERT [1] (Bidirectional Encoder Representation Transformers) achieves that coding in eleven tasks of which intent and slot prediction [1,2]. One way of using such a language model consists in connecting a simple neural network (NN) architecture, such as a linear neural network (LNN, a single layer) and fine-tuning all language model and NN parameters end-to-end [1]; however, other NN architectures may be used such as a recurrent NN (RNN) [3], a long short-term memory (LSTM) [4], or a bidirectional long short-term memory (BiLSTM) [5]. The success of BERT led the scientific community to extend it to other languages than English; in French, this extension generated two models: CamemBERT [6] and FlauBERT [7].

In the literature, few researchers have focused on intent and slot prediction with language models, a chatbot, and a French corpus. Between 2018 and 2020, for intent and slot prediction in the medical domain, Neuraz et al. used ELMo and FastText, two word embedding methods for representing sequences of words as corresponding sequences of vectors. Their three papers [8–10] showed a clear superiority of ELMo over FastText in terms of F1 score. Later, in 2020, CamemBERT and FlauBERT started being used for the same purpose (the present work) and others too. As the latter two language models are based on Transformers (one of the most powerful neural architectures today), they became the current references in NLP. Concomitantly, FlauBERT performed better than FastText in “diverse NLP tasks (text classification, paraphrasing, natural language inference, parsing, word sense disambiguation)” targeting of various content texts [7] and CamemBERT performed better than ELMo in processing sentiment analysis from texts from very various sources [11].

Since 2020, in the medical field, Anastasiadou et al. [12] compared BERT against a support vector machine (SVM) and a conditional random field (CRF) in a chatbot for diabetes management and, in 2021, Lei et al. [13] used BERT for COVID-19 patient monitoring. In both cases, the chatbots were domain-specific (diabetes and COVID-19). However, up to now, neither CamemBERT nor FlauBERT has been previously used in a medical chatbot on a much more extended medical corpus.

This study aims to compare CamemBERT and FlauBERT abilities to extract intents and slots from a French medical corpus and determine the best language model/neural network architecture combination able to help patients and health care professionals.

2. Materials and methods

2.1. The data

The present study used the French CAS corpus [14], a benchmark among French medical corpuses. CAS includes thousands of clinical reports extracted from the specialized literature. From that corpus, the study analyzed a sample of 1133 sentences. From each sentence, a single intent and one or several slots were manually tagged according to the taxonomy detailed in Table 2.

The slots were tagged using the IOB format. In that format, prefix ‘B-’ indicates a word at the beginning of a slot, ‘I-’ a word in the middle or the end of a slot. Prefix ‘O’ was used to tag words that do not belong to slots. Finally, ‘</s>’ indicates the end of a sentence. For example, the sentence shown in Table 1 (‘J'ai eu de la tachycardie hier soir </s>’) would be tagged: O O O O O B-TypePresent B-Time I-Time Symptom.

Table 1
Formal representation of a sentence by intent and slots.

Speech sentence	Intent	Slots
J'ai eu de la tachycardie hier soir[a]	Symptom	Type present (*tachycardie*) Time (*hier soir*)

[a] I had tachycardia last night.

Table 2
Taxonomy and dataset description.

Intents and slots	Description	Frequency (%)
Patient		192 (17%)
Weight	Patient's weight	44 (14%)
Height	Patient's height	40 (12%)
Age	Patient's age	172 (53%)
Person	Patient's name	69 (21%)
Symptom		203 (18%)
Type present	Symptom description	377 (76%)
Time	Date of symptom onset	121 (24%)
Exam		566 (50%)
Type present	Past medical examination	639 (100%)
Risk		172 (15%)
Type present	Presence of risk factor	218 (65%)
Type absent	Absence of risk factor	116 (35%)

2.2. The proposed approaches

Intent and slot prediction was carried out in two steps (Fig. 1). First, the language model encoded each word of a sentence with an embedding layer and that encoding was contextualized with Transformer Encoder layers. Second, using the contextualized embeddings generated by the language model, a neural network architecture predicted intents and slots.

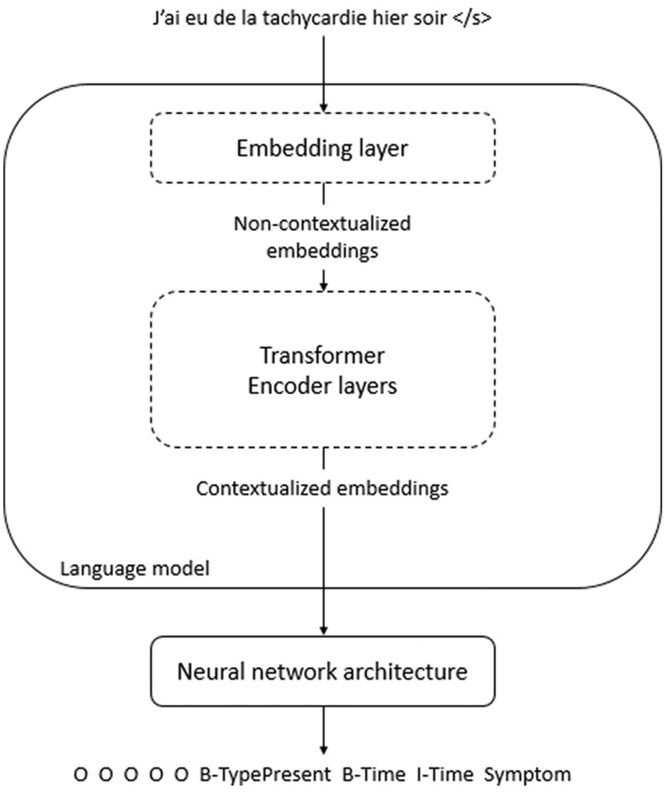
J'ai eu de la tachycardie hier soir </s>
Embedding layer
Non-contextualized embeddings
Transformer Encoder layers
Contextualized embeddings
Language model
Neural network architecture
O O O O O B-TypePresent B-Time I-Time Symptom

Fig. 1. Intent and slot prediction using a language model. At this figure top, tag </s> indicates the end of the sentence under analysis. First, the language model encodes the sentence non-contextually (with the embedding layer) then contextually (with the Transformer Encoder layers). Finally, the neural network architecture predicts the intent and the slots using the previously calculated contextualized embeddings. The slots are in IOB format: prefix ‘B-’ indicates a word at the beginning of a slot, ‘I-’ a word in the middle or the end of a slot, and ‘O’ a word that does not belong to a slot.

2.2.1. The language models

Language models analyze the text in segments of one or more sentences. In the present study, it is important to underline that each segment corresponded to a single sentence. Each word (sometimes part of the same slot; e.g., slot 'hier soir' derives from 'hier' (B-Time) and 'soir' (I-Time)) was indexed to a vocabulary composed of words and subwords (also called 'tokens'). A tokenizer allowed managing unknown words by cutting them up and then associating them with vocabulary tokens. The embedding layer returned, for each token, a non-contextualized embedding that corresponds to the sum of three different embedding (Fig. 2):

- A token embedding resulting from a linear projection of the tokens indexes.
- A position embedding resulting from a linear projection of tokens' positions in the segment (here, = sentence).
- A segment embedding resulting from a linear projection of the belonging of various tokens of the same sentence. Here, all tokens of a given sentence had the same segment embedding because each segment corresponded to a single sentence.

The non-contextualized embedding resulting from the embedding layer passed then through a succession of twelve Transformer Encoder layers [15] (Fig. 1). Each layer took as input the previous single output to refine the contextualization using a twelve-headed self-attention mechanism. In each of the twelve heads, the self-attention mechanism allowed each token of the sentence to find out other token needed to refine the contextualization. Finally, the language model returned a contextualized embedding for each token of the sentence.

Here, two language models were compared: FlauBERT and CamemBERT. These two models have similar structures but differ in a number of points: i) the tokenizer (Byte Pair Encoding [16] vs. SentencePiece [17], respectively); ii) the vocabulary size (50,000 vs. 32,000, resp.); iii) the number of parameter (138M vs. 110M, resp.); and, iv) the size of the training dataset (71 GB vs. 138 GB, resp.).

2.2.2. The neural network architectures

From a contextualized embedding and for each token, the NN architecture infers the logarithm of a probability distribution for each potential intent and slot using a LogSoftmax activation function. Finally, a CRF [18] was attached at the top of every NN architecture to predict the most likely intent or slot while preserving the IOB format.

Here, the results of using four NN architectures (each) associated with a CRF were compared:

- a Linear NN (LNN) that processes linearly each contextualized embedding.
- a Recurrent NN (RNN) that processes each contextualized embedding taking into account the previous one.
- a Long Short-Term Memory (LSTM) that processes each contextualized embedding taking into account all previous ones.
- a Bidirectional Long Short-Term Memory (BiLSTM) that processes each contextualized embedding taking into account all previous and subsequent ones.

2.3. The training parameters

To train jointly all language models and NN parameters end-to-end, the number of epochs (i.e., number of times the corpus is explored during training) was set to ten as determined by the convergence value of the negative log-likelihood. AdamW learning algorithm [19] was used for training with a learning rate initially set to 2e-5 (first epoch) but that decreased linearly until it came to 0 (last epoch). A gradient clipping [20] of 1.0 was used to reduce gradient-exploding effects; otherwise said, the gradient was scaled down whenever its norm exceeded 1.0 to avoid too large gradients. Finally, every NN architecture has only one hidden layer.

In this work, we used Python 3.8.2 as programming language and the following packages:

- Torchtext 0.9.1 to load and tokenize the CAS corpus.
- Transformers 3.1.0 from HuggingFace to apply CamemBERT and FlauBERT.
- PyTorch 1.8.1 to deal with the NN architecture, the CRF, and model training.

With an NVIDIA Graphics processing Unit of 16 GB, the processing time for the downstream task was about 20 ms per sentence and language model.

2.4. The evaluation criteria

Intent and slot predictions were separately evaluated with Macro F1 scores. A standard confusion matrix was calculated for intents. For slots, a 'true positive' slot was one whose every token was correctly predicted, a 'false positive' one whose not all tokens were correctly predicted, and a 'false negative' one that was not predicted at all. Combinations of intent and slot predictions were used to analyze a joint performance.

The evaluation used a bootstrap method [21]. That method consisted in drawing randomly sentences from the CAS corpus –with replacement

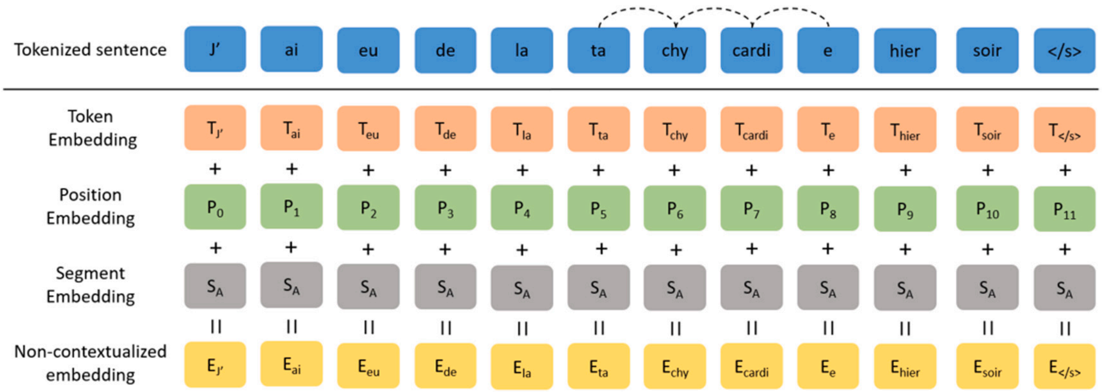
Tokenized sentence
J'
ai
eu
de
la
ta
chy
cardi
e
hier
soir
</s>
Token Embedding
$T_{J'}$ T_{ai} T_{eu} T_{de} T_{la} T_{ta} T_{chy} T_{cardi} T_{e} T_{hier} T_{soir} $T_{</s>}$
Position Embedding
P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}
Segment Embedding
S_A
Non-contextualized embedding
$E_{J'}$ E_{ai} E_{eu} E_{de} E_{la} E_{ta} E_{chy} E_{cardi} E_{e} E_{hier} E_{soir} $E_{</s>}$

Fig. 2. Representation of the embedding layers. The tokenized sentence shows the tokens of the sentence (here, = segment) obtained with the tokenizer submitted to analysis. In this sentence, 'tachycardie' was cut into four tokens. The token embedding is a linear projection of the indexes given to the tokens of the sentence. The position embedding is a linear projection of the position of each token in the sentence. The segment embedding is a linear projection of the belonging of various tokens to the same sentence. Finally, the embedding layer returns a non-contextualized embedding that corresponds to the sum of the embeddings of the token, the position, and the segment.

to keep constant the size of the training set. Thus, the training set included 1133 sentences (just as the CAS corpus). Sentences not drawn constituted the test set. Theoretically, about 37% of the sentences of CAS corpus remained in the test set (nearly 419 sentences). This bootstrapping was slot-stratified and fifty bootstrap iterations were run to obtain means and standard deviations for the Macro F1 score.

3. Results

3.1. Performance of the language models

According to the Macro F1 scores, FlauBERT performed better than CamemBERT regardless of the NN architecture used (Table 3). Regarding intent prediction, FlauBERT and CamemBERT scores were rather very close (differences less than 0.02 according to the architecture). Regarding slot prediction, FlauBERT and CamemBERT scores were less close (differences less than 0.2 according to the architecture).

Regarding the combinations of intent and slot predictions, FlauBERT performed better than CamemBERT regardless of the NN architecture used (Table 4). FlauBERT and CamemBERT predicted correctly the intent and every slot with NN architecture. However, both language models were wrong about at least one slot in proportions very close to 0.19 with the former and 0.25 with the latter whatever the NN architecture. Anyway, the proportion of wrongly predicted intents never exceeded 0.33.

3.2. Performance of the NN architectures

According to the Macro F1 scores, the NN architectures had heterogeneous performance levels that depended on the language model used (Table 3). Regarding intent prediction, LNN, RNN, LSTM, and BiLSTM architectures had very close and not significantly different Macro F1 scores with either FlauBERT (circa 0.972) or CamemBERT (circa 0.957). Regarding slot prediction with FlauBERT, Macro F1 scores with RNN, LSTM, and BiLSTM architectures were slightly higher (though not significantly different) than with the LNN. Regarding slot prediction with CamemBERT, Macro F1 scores with a LNN or a RNN architecture were higher (though not significantly different) than with a LSTM or a BiLSTM architecture.

According to the combinations of intent and slot predictions, the NN architectures had heterogeneous performance levels that depended on the language model used (Table 4). All architectures (LNN, RNN, LSTM, and BiLSTM) predicted correctly the intent and every slot in proportions close to 0.785 with FlauBERT and to 0.710 with CamemBERT (the differences were not significant). LNN, RNN, LSTM, and BiLSTM architectures predicted correctly the intent but were wrong about at least one slot in proportions close to 0.190 with FlauBERT and 0.240 with CamemBERT (differences not significant). The proportion of wrongly predicted intents never exceeded 0.033.

Table 3
Macro F1 scores for intent and slot predictions.

Language model and architecture	Intent F1 score	Slot F1 score
CamemBERT		
LNN architecture	0.957	0.836[a]
RNN architecture	0.955[a]	0.838[a]
LSTM architecture	0.953	0.805[a]
BiLSTM architecture	0.952[a]	0.814[a]
FlauBERT		
LNN architecture	0.975	0.879[a]
RNN architecture	0.972	0.883
LSTM architecture	0.970	0.884[a]
BiLSTM architecture	0.972	0.883[a]

LNN: linear neural network. RNN: recurrent neural network. LSTM: long short-term memory. BiLSTM: bidirectional long short-term memory.

[a] Standard deviation range: 0.010–0.015. All other standard deviations range between 0.006 and 0.010.

Table 4
Distribution of intent and slot prediction combinations.

Language model and architecture	Intent true		Intent false	
	Slot true	Slot false	Slot true	Slot false
CamemBERT				
LNN architecture	0.714[a]	0.239[a]	0.014	0.033
RNN architecture	0.719[a]	0.236[a]	0.015	0.030
LSTM architecture	0.585[a]	0.264[a]	0.017	0.033
BiLSTM architecture	0.695[a]	0.258	0.014	0.033
FlauBERT				
LNN architecture	0.779[b]	0.197[b]	0.009	0.015
RNN architecture	0.786[b]	0.189[b]	0.010	0.015
LSTM architecture	0.787[a]	0.186[a]	0.010	0.017
BiLSTM architecture	0.785[a]	0.185[a]	0.012	0.018

All other standard deviations range between 0.006 and 0.010. LNN: linear neural network. RNN: recurrent neural network. LSTM: long short-term memory. BiLSTM: bidirectional long short-term memory.

[a] Standard deviation range: 0.015–0.020.

[b] Standard deviation range: 0.020–0.25.

4. Discussion

Given the above-shown results, FlauBERT performed better than CamemBERT regardless of the NN architecture used, whereas the NN architectures performed unequally depending on the language model. RNN, LSTM, and BiLSTM architectures outperformed slightly the LNN with FlauBERT but the LNN and RNN architectures outperformed slightly LSTM and BiLSTM architectures with CamemBERT. Undeniably, the best combinations were those that used FlauBERT.

One plausible explanation for FlauBERT better performance would be its extra 28M parameters vs. CamemBERT. Indeed, Brow et al. [22] have recently demonstrated that the number of parameters in a language model has a strong impact and that the higher is the number of parameters, the better are the results. Another explanation would be the 18,000-token vocabulary difference that make FlauBERT able to better deal with the input words and provide better contextual representations.

The NN architectures were difficult to compare given the non-significant differences in most Macro F1 scores. However, it seemed that, with FlauBERT, the architectures that compute the tokens and take into account the rest of the sentence (i.e., RNN, LSTM, and BiLSTM) performed better than the LNN. With CamemBERT, the opposite was seen: LSTM and BiLSTM architectures performed poorly vs. the LNN and the RNN architectures. Thus, for the moment, concluding about architecture performance seems very difficult; it requires another study on much more data (say, 10 times more).

One merit of the work is the addition to the current literature new results stemming from the use and comparisons of performance between CamemBERT and FlauBERT within the context of a medical chatbot in French language. Up to now, FastText and ELMo were the only language models used this way [8–10]; they showed much lower performance than those of CamemBERT and FlauBERT in most NLP tasks. The performances of the above-cited four word embedding methods are certainly worth being compared on the same medical corpus in a future dedicated work. This will be a natural and interesting extension of the present work.

Although CamemBERT and FlauBERT were used here with French language, it seems highly probable that the same kind of study could be conducted with other languages that have similar sentence structures (syntaxes) and may undergo similar contextual embeddings; e.g., Spanish, Italian, Portuguese. Moreover, the CAS corpus used [14] included nearly 200 medical case reports in nearly all medical specialties. This wide coverage is a non-negligible asset versus contexts with a single specialty or domain (e.g., diabetes [12] or COVID-19 [13]). A second merit of the work is the prediction of intents in addition to slots.

Intent prediction helps slot prediction and allows a better organization of the information contained in a sentence. A third merit is the manual labeling of the corpus. Manual intent and slot labeling provides a quality corpus analysis that accounts for the accuracy and sensitivity of the medical language. Furthermore, manual labeling allows adapting accurately different performant language models to the medical chatbot; thus, extracting relevant information for the health care professional. However, one inconvenience of manual labeling is that it may be tedious and time-consuming (depending on the corpus size). For the present work, the manual labeling of the whole corpus required nearly two weeks full-time work.

5. Conclusion

In this comparison of intent and slot prediction between CamemBERT and FlauBERT fine-tuned with different NN architectures in a medical chatbot for French-speaking patients, i) FlauBERT achieved a better performance regardless of the NN architecture; and, ii) the most complex architectures did not significantly outperform the LNN which seemed to be the most reliable. Thus, for a French medical chatbot, we would recommend FlauBERT with a LNN.

Declaration of competing interest

Authors CB, AB, EF, and TG are employed by Everteam Software. Authors FJ, BW, and PR have no interests to declare.

Acknowledgments

The authors are grateful to Jean Iwaz (Hospices Civils de Lyon) for augmenting, editing, proofreading, and formatting the latest versions of the manuscript.

Funding

This work was supported by Association Nationale de la Recherche et de la Technologie (ANRT) [grant number 2019/1374]. The sponsor had no role in the study design; the collection, analysis, and interpretation of the data; the writing of the report; and the decision to submit the article for publication.

References

[1] Devlin J, Chang M-W, Lee K, Toutanova K, BERT. Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference North American Chapter of the Association for Computational Linguistics: human language technologies. 1; 2019. p. 4171–86.

[2] Chen Q, Zhuo Z, Wang W. BERT for joint intent classification and slot filling. 2019. http://arxiv.org/abs/1902.10909.

[3] Elman JL. Finding structure in time. Cognit Sci 1990;14:179–211. https://doi.org/10.1016/0364-0213(90)90002-E.

[4] Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schimdhuber J. LSTM: a search space odyssey. IEEE Trans Neural Netw Learn Syst 2017;28:2222–32. https://doi.org/10.1109/TNNLS.2016.2582924.

[5] Schuster M, Paliwal K. Bidirectional recurrent neural networks. IEEE Trans Signal Process 1997;45:2673–81. https://doi.org/10.1109/78.650093.

[6] Martin L, Muller B, Suarez PJO, Dupont Y, Romary L, de la Clergerie V, Seddah D, Sagot B. CamemBERT: a tasty french language model. In: Proceedings of the 58th annual meeting of the Association for Computational Linguistics; 2020. p. 7203–19. https://doi.org/10.18653/v1/2020.acl-main.645.

[7] Le H, Vial L, Frej J, Segonne V, Coavoux M, Lecouteux B, Allauzen A, Crabbé B, Besacier L, Schwab D. FlauBERT: unsupervised language model pre-training for french. In: Proceedings of the 12th Language Resources and Evaluation Conference; 2020. p. 2479–90.

[8] Neuraz A, Looten V, Rance B, Daniel N, Garcelon N, Llanos LC, Burgun A, Rosset S. Do you need embeddings trained on a massive specialized corpus for your clinical natural language processing task? Stud Health Technol Inform 2019;264:1558–9. https://doi.org/10.3233/SHTI190533.

[9] Neuraz A, Rance B, Garcelon N, Llanos LC, Burgun A, S. Rosset S.. The impact of specialized corpora for word embeddings in natural language understanding. Stud Health Technol Inform 2020;270:432–6. https://doi.org/10.3233/SHTI200197.

[10] Neuraz A, Llanos LC, Burgun A, Rosset S. Natural language understanding for task oriented dialog in the biomedical domain in a low resources context. In: Machine Learning for Health (ML4H) Workshop at NeurIPS; 2018. https://arxiv.org/pdf/1811.09417.pdf.

[11] Habbat N, Anoun H, Hassouni L. LSTM-CNN deep learning model for french online product reviews classification. In: Saidi R, Bhiri B El, Maleh Y, Mosallam A, Essaaidi M, editors. International conference on advanced technologies for humanity (ICATH 2021). 110; 2022. p. 228–40. https://doi.org/10.1007/978-3-030-94188-8_22.

[12] Anastasiadou M, Alexiadis A, Polychronidou E, Votis K, Tzovaras D. A prototype educational virtual assistant for diabetes management. In: IEEE 20th international conference on bioinformatics and bioengineering (BIBE); 2020. p. 999–1004. https://doi.org/10.1109/BIBE50027.2020.00169.

[13] Lei H, Lu W, Ji A, Bertram E, Gao P, Jiang X, Barman A. Covid-19 smart chatbot prototype for patient monitoring. 2021. https://arxiv.org/abs/2103.06816.

[14] Grabar N, Claveau V, Dalloux C. CAS: French corpus with clinical cases. In: Proceedings of the ninth international workshop on health text mining and information analysis. Association for Computational Linguistics; 2018. p. 122–8. https://doi.org/10.18653/v1/W18-5614.

[15] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: The 31st conference on neural information processing systems; 2017. p. 1–11.

[16] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units. In: Proceedings of the 54th annual meeting of the association for computational linguistics; 2016. p. 1715–25. https://doi.org/10.18653/v1/P16-1162.

[17] Kudo T, Richardson J. In: Sentence piece: a simple and language independent subword tokenizer and detokenizer for neural text processing. Association for Computational Linguistics; 2018. p. 66–71. https://doi.org/10.18653/v1/D18-2012.

[18] Lafferty J, McCallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML '01 proceedings of the 18th international conference on machine learning; 2001. p. 282–9.

[19] Loshchilov I, Hutter F. Decoupled weight decay regularization. In: 7th international conference on learning representations; 2019.

[20] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. In: Proceedings of the 30th international conference on international conference on machine learning. 28; 2013. p. 1310–8.

[21] Efron B. Bootstrap methods: another look at the jackknife. Ann Stat 1979;7:1–26. https://doi.org/10.1214/aos/1176344552.

[22] 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.F. Balcan H. Lin , Editors. Advances in neural information processing systems 33, Annual conference on neural information processing systems (NeurIPS 2020). https://arxiv.org/pdf/2005.14165.pdf.