ISSN 2224-087X. Електроніка та інформаційні технології. 2023. Випуск 23. С. 46–58
Electronics and information technologies. 2023. Issue 23. P. 46–58

УДК 004 DOI: https://doi.org/10.30970/eli.23.5

NAMED ENTITY RECOGNITION USING OPENAI GPT SERIES MODELS

B. Pavlyshenko, I. Drozdov

Ivan Franko National University of Lviv
50 Drahomanova St., UA-79005 Lviv, Ukraine
bohdan.pavlyshenko@lnu.edu.ua, ihor.drozdov@lnu.edu.ua

The amount of information grew dramatically over all available sources. One of the most important parts of it is textual data, so Natural Language Processing is one of the most important areas of research. A growing amount of information demands more sophisticated and effective models and approaches to be introduced. Named entity recognition is a key part of text processing and plays an important role in text understanding, automatic text summarization, translation, etc. A wide range of different approaches were used for named entity recognition, however, the introduction of the transformers architecture with self-attention mechanism made a significant impact on current approaches to Natural Language Processing tasks in general. Most tasks are currently leveraging transformers as a state-of-the-art approach. Meanwhile, simpler transformer architecture in comparison with others grants the possibility of large language models with a huge number of parameters like GPT-3.

The main purpose of this article is to investigate how effectively OpenAI GPT series models could recognize named entities in English and Ukrainian texts. The research was based on the CoNLL 2003 dataset one of the most used for such kind of research and the lang-uk team labeled dataset. Due to known possibilities for GPT series models to be more effective with few-shot learning examples, experiments were built with zero, one, and three shots. Moreover, experiments were performed for whole articles and sentence by sentence from the same article to compare results. Different prompts were investigated, and one was chosen for the whole experiment. The estimation of the results was based on the F1 score and specifics of the results. Results demonstrate the overall great performance of the most recent models and the increase in performance from older to newer models. Furthermore, our findings indicate that there is still room for improvement and investigation.

Keywords: Named entity recognition, natural language processing, GPT, OpenAI

Introduction

Named Entity Recognition (NER) is one of the most important tasks from the Natural Language Processing (NLP) area and aims to recognize specific predefined types such as person, organization, location, etc. As noted in [1],[2], NER is not only a standalone task, but it also has a significant role in other Natural language processing tasks like text understanding, automatic text summarization, translation, etc. Named Entities Recognition as a separate sub-task of the NLP was separated during the sixth Message Understanding Conference (MUC-6) [3] as the task of identifying names of organizations, people, locations, currency, time, etc.

Starting from MUC-6 interest to NER significantly increased. During the past 30 years, multiple approaches were used for NER, but all of them are possible to split into four main directions: 1) Rule-based approach; 2) Unsupervised learning approaches; 3) Feature-based supervised learning approaches; 4) Deep learning approaches.

In the last decade, the amount of information growth significantly along with the performance of the chips, used to train complex models, as a result, deep learning approaches become dominant for most machine learning tasks, including NLP and NER as its part. The deep learning approach as a concept, consists of multiple processing layers, which usually represent different levels of abstraction [4]. This allows for deep learning models to automatically identify key features from raw data required for classification or detection. Key strengths, which can be achieved by using deep learning approaches to NER: benefits from non-linear mappings between input values and output ones, deep learning models can effectively detect key features from raw data without good engineering skill and domain expertise.

During the last decade, hundreds of different approaches were introduced, based on [1-2,5-6] schematic architecture could be defined in 3 subjunctive layers: 1) Distributed representations for input(word embeddings, character-level embeddings, POS tag, Gazetteer, etc.); 2) Context encoder(CNN, RNN, Language model, Transformer, etc.); 3) Tag decoder (Softmax, CRF, RNN, Point network, etc.). Most architectures are typically based on complex Convolutional or recurrent neural networks, which include an encoder and a decoder. Vaswani et al. [7] proposed a new simple network, named the Transformer, which is based on self-attention mechanisms and doesn't use recurrence and convolutions at all. This model shows superior quality and possibilities in parallelization. Based on Transformers, BERT (bi-directional transformers for language understanding) was introduced [8]. Current state of the art for the most popular datasets uses Transformers [9,10].

Despite demonstrating great results, many NLP tasks still depend on fine-tuning for a specific task. As an important research direction, Generative Pre-trained Transformer (GPT) model was introduced by the OpenAI research team. Brown T. et al. [11] showed how the model can adopt with zero or few-shot training. GPT-3 model has 175B parameters and 96 layers. The model was pre-trained on a huge text corpus and could be used for any kind of NLP task without any preparation or with few-shots examples. Based on research at [12,13], these models are far behind the state of the art in NER tasks, but the generative approach have own benefits, and improvement progress is significant during the last couple of years.

During the last year, big attention to models from GPT series was involved. Despite this, research about its possibilities is still limited. The goal of this article is to investigate the possibilities of GPT series models to effectively recognize named entities in plain text for English and Ukrainian texts.

Methods and materials

During the research, two datasets were used: CoNLL 2003 [14] as one of the most popular datasets for NER model estimation, and the Ukrainian dataset marked by the lang-uk team [15] based on the Brown corpus of Ukrainian texts. Table 1 contains information about datasets. For CoNLL usual split on train, development, and test sets was used [14], for the lang-uk dataset, a random split with weights 8:1:1. Train and development sets were used for model training and validation respectively, and the test set was used for model estimation.

For experiments, five GPT series models were selected, and all of them represent evolution process and capabilities [16]:

- Text-ada-001 – capable for very simple tasks, very fast, and the cheapest.
- Text-babbage-001 – capable for straightforward tasks, fast and cheap.
- Text-curie-001 – good capability, faster and cheaper than text-davinci-003.
- Text-davinci-003 – have great capability, and quality, but is slower than others and expensive.
- Gpt-3.5-turbo – the most capable and 10x cheaper than text-davinci-003 model. Optimized for a chat.

Table 1. Information about used datasets.

CoNLL 2003 dataset, English							
	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	946	14,987	203,621	7140	3438	6321	6600
Development set	216	3,466	51,362	1837	922	1341	1842
Test set	231	3,684	46,435	1668	702	1661	1617
Lang-uk dataset, Ukrainian							
	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	210	---	187,097	1,281	563	643	3,431
Development set	20	---	23,494	163	62	59	513
Test Set	29	---	26,883	170	35	78	441

GPT series language models developed by OpenAI for generating natural language text based on human-like requests with task descriptions. Because these models understand requirements from the prompt using in-context learning, different structure, phrases, paragraphs, and format of expected results - everything could change significantly final model response. Moreover, as Large Language models have shown promising results with few-shot on-context learning [11], for experiments prompts were decided to use zero, one, and three samples per prompt. In brief, one sample is input text and named entities, which this text contains (Figures 1-2). Before the experiment, multiple different prompt formats were used, and choose one with the best results on three sentences from the training dataset. Fig. 1 demonstrates the final prompt, which was used during experiments, and Fig. 2 shows an example of a one-shot prompt. Even though some experiments with prompts were made, comprehensive research on the prompt request was out of the scope of this paper. However, it is an important direction for further research.

```
You are highly-precise named entity recognition system.
Please find Named entities with the next types:
ORG(Organisation), PER(Person), LOC(Location), MISC(Miscellaneous).
If entity exists more than one time, output as many as you find in text.
Entities should have the same order as in text.
Use this output format:
[    T:E,   T:E,  ...] Where T - entity type, E - entity in text
{% if examples is defined and examples|length > 0 -%}
Examples:
{% for sentence, label in examples %}
Input: {{ sentence }}
Output: [{{ label }}]
{% endfor %}{% endif -%}
Input:{{text_input}}
Output:
```

Fig. 1. Template for request prompt.

```
You are highly-precise named entity recognition system.
Please find Named entities with the next types: ORG(Organisation), PER(Person), LOC(Location),
MISC(Miscellaneous).
If entity exists more than one time, output as many as you find in text. Entities should have the same
order as in text.
Use this output format: [ T:E, T:E, ... ] Where T - entity type, E - entity in text
Examples:
Input: Police said the 111 passengers and six crew on board the ferry Trident Seven, owned by France 's
Emeraud line, were rescued by a variety of private and commercial boats after fire broke out in the
engine room soon after it left port.- Analyst Alexander Paris said he expected consistent 20 percent
earnings growth after an estimated gain of 18 percent for 1996.
Output: [ MISC:Trident Seven, LOC:France, ORG:Emeraud, PER:Alexander Paris ]

Input: The acquisition will beef up Markham, Ontario-based Magna 's North American car and truck
seating business, allowing it to better compete with Johnson Controls Inc and Lear Corp.
Output: [ LOC:Markham, ORG:Magna, ORG: Johnson Controls Inc, ORG: Lear Corp ]
```

Fig. 2. Example of the request for a one-shot prompt.

Due to limited resources available, CoNLL dataset experiments were built in the next way:

- 2 experiments were executed.
- For each experiment, 25 articles were selected randomly from the test dataset.
- Each chosen article should have more than the average sentences count per test dataset and the named entities count should be more than the average per dataset.
- For sampling, the same sentences were used for all prompts in one experiment. From one to three sentences were randomly selected from the train dataset per example. The total length should have at least 65 tokens.

- Experiments were executed in two different ways: full article in one prompt and prompt per each sentence in the article with zero, one, and three samples.
- All five GPT series models were used.

Based on lang-uk dataset characteristics, the experiment was built in a more limited way:

- Only one experiment was executed.
- Based on the capabilities of the models, only text-davinci-003 and gpt-3.5-turbo were used.
- For the experiment, from the test dataset were selected all articles with a number of named entities at least 10.
- Experiments were executed only per each sentence prompt with zero, one, and three samples. Articles are too big and exceed the maximal prompt size.
- Only the most capable GPT series models wire used: text-davinci-003, gpt-3.5-turbo.

The experiments were performed using OpenAI API [16]. For simplification of the experiment workflow, additional libraries like Promptify [17] were used. Promptify allows to make requests to OpenAI API with a simple wrapper and simplifies the overall process.

In addition, RoBERTa model for CoNLL dataset was trained. For the training process training and development sets (Table 1) were used. Firstly, spaCy framework [18] was chosen as it provides different pipelines for NLP tasks and a simple pipeline for custom models training and usage. The training process with spaCy is based on a pre-defined config file with possibility to customize for task needs. In this case, "roberta-base" model from spaCy framework was chosen and trained for efficiency using GPU on MacBook Pro 2018 and Radeon Pro 560X video card. Overall training process progress is demonstrated in Figure 3. As GPT series models experiments were performed on a limited part of the CoNLL's test dataset, trained RoBERTa model was estimated on the same articles and with the same measurement system.

```
============================= Training pipeline =============================
ℹ Pipeline: ['transformer', 'ner']
ℹ Initial learn rate: 0.0
```

E	#	LOSS TRANS...	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	1144.66	1393.23	2.11	1.18	10.23	0.02
1	200	41503.22	51979.61	90.85	90.44	91.27	0.91
2	400	2562.84	4023.01	92.96	92.69	93.23	0.93
3	600	1647.78	2538.25	93.27	93.01	93.54	0.93
4	800	1134.46	1640.94	94.78	95.12	94.45	0.95
5	1000	705.86	1173.42	95.33	95.11	95.56	0.95
6	1200	519.46	860.49	95.74	95.71	95.78	0.96
7	1400	498.65	810.95	95.56	95.68	95.44	0.96
8	1600	443.22	591.43	96.01	95.98	96.05	0.96
9	1800	331.81	469.37	96.15	96.44	95.86	0.96
10	2000	292.17	456.93	95.45	95.21	95.69	0.95
11	2200	391.73	542.13	95.14	95.15	95.14	0.95
12	2400	212.83	323.72	95.73	95.53	95.93	0.96
13	2600	158.21	285.46	95.92	95.79	96.06	0.96
15	2800	176.37	252.48	95.16	94.61	95.71	0.95
16	3000	185.95	264.56	95.63	95.39	95.86	0.96
17	3200	145.43	193.98	95.96	95.67	96.25	0.96
18	3400	104.90	136.57	95.55	95.23	95.86	0.96

Fig. 3. RoBERTa model training process

Measurement system.

Large language models like GPT series were developed for generating human-like language text. Consequently, it is a challenging task to receive a sequence of the named entities in the order they appear in the text and with the same quantity. In fact, multiple different approaches how to estimate NER model exists nowadays. Finally, for measurement was used criteria: “named entity is detected and with the right label”. Based on these criteria, F1 score was defined as:

TP_c= all detected entities with label C and has the same class in the dataset.

FP_c= all detected entities with label C, but the entity doesn’t exist in the original dataset or has another class.

FN_c= all entities with label C, which is not detected from the original dataset OR entity is detected with other than label C, but in the original dataset has class C.

During calculations, every unique combination <entity, label> is used only one time.

Measurement was made after each prompt: for each sentence or document, a comparison of the results with the original named entities was made.

Results and analysis.

In brief, after making experiments, some expected results were received, but also some not expected findings were detected.

CoNLL dataset. Table 2 and Table 3 contain experiment results with full article prompts and per-sentence prompts respectively. As two experiments were performed, results in each cell are represented as average value ± maximum difference from average.

Table 2. CoNLL. Scoring for experiments execution for full article prompts. The biggest value in each block per shot is bolded.

algorithm	PER F1	ORG F1	LOC F1	MISC F1	Precision	Recall	F1
spaCy-RoBERTa	96.77±0.23	90.18±1.44	89.72±0.96	79.80±0.54	90.03±1.81	91.62±0.80	90.81±1.31
0-shot learning							
text-ada-001	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
text-babbage-001	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
text-curie-001	0.85±0.85	0.68±0.68	0.00±0.00	0.00±0.00	75.00±25.00	0.27±0.11	0.52±0.23
text-davinci-003	**82.53±1.92**	**62.13±2.13**	**68.30±3.13**	**5.20±2.21**	**81.60±1.42**	**54.11±1.45**	**65.07±1.50**
gpt-3.5-turbo	20.03±10.96	18.34±4.95	15.20±4.09	3.82±0.23	21.43±8.78	13.97±5.98	16.91±7.12
1-shot learning							
text-ada-001	2.79±2.79	3.60±3.60	7.05±5.85	0.80±0.80	23.52±8.13	1.98±1.73	3.56±3.06
text-babbage-001	1.70±1.70	0.68±0.68	2.94±1.77	0.00±0.00	45.10±21.57	0.70±0.19	1.37±0.38
text-curie-001	7.20±6.08	5.33±5.33	22.62±2.00	1.48±1.48	39.04±9.04	5.41±2.00	9.47±3.35
text-davinci-003	65.72±15.51	56.27±6.27	**57.42±5.82**	4.49±1.53	78.48±7.44	40.68±4.44	53.58±5.59
gpt-3.5-turbo	**76.96±2.23**	**57.94±2.28**	53.52±0.20	**4.64±3.17**	**86.44±1.09**	**44.64±4.10**	**58.78±3.83**

3-shot learning							
text-ada-001	2.21±0.53	7.98±0.10	9.77±0.93	3.54±0.12	19.32±0.23	3.19±0.08	5.48±0.14
text-babbage-001	7.74±4.57	0.00±0.00	8.78±4.05	5.72±4.45	26.81±12.14	3.43±2.04	6.07±3.53
text-curie-001	24.82±13.43	17.21±9.39	36.95±10.02	11.59±2.77	38.53±13.35	17.02±7.54	23.54±9.79
text-davinci-003	76.27±2.36	**56.01±0.55**	**68.52±3.56**	23.63±2.07	**84.01±2.03**	49.67±3.23	**62.41±3.11**
gpt-3.5-turbo	**76.72±5.78**	51.97±3.23	66.13±2.64	**26.63±0.31**	79.02±1.24	**50.50±4.93**	61.52±4.05

Table 3. CoNLL. Scoring for experiments execution for sentence prompts. The biggest value in each block per shot is bolded.

algorithm	PER F1	ORG F1	LOC F1	MISC F1	Precision	Recall	F1
spaCy-RoBERTa	95.25±0.88	89.94±0.56	89.75±1.25	85.40±0.38	90.32±1.19	91.34±0.78	90.83±0.98
0-shot learning							
text-ada-001	0.54±0.10	0.00±0.00	0.00±0.00	0.00±0.00	11.31±2.98	0.08±0.01	0.18±0.01
text-babbage-001	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
text-curie-001	0.92±0.32	0.73±0.73	6.26±1.41	0.00±0.00	11.95±4.57	1.17±0.32	2.13±0.61
text-davinci-003	**87.77±2.75**	**54.54±2.00**	**53.64±2.79**	5.65±0.92	**46.17±5.76**	**65.28±2.77**	**54.00±4.91**
gpt-3.5-turbo	56.42±5.04	34.06±4.12	42.47±1.80	**12.82±0.33**	35.09±1.83	46.25±0.68	39.86±0.93
1-shot learning							
text-ada-001	6.87±6.87	6.73±3.29	10.18±5.82	3.30±0.51	11.14±1.20	5.66±4.17	6.77±4.17
text-babbage-001	0.00±0.00	0.00±0.00	2.54±1.27	2.94±1.57	9.96±1.63	0.71±0.05	1.32±0.06
text-curie-001	11.58±4.54	1.60±1.60	31.83±8.32	5.75±3.02	20.19±1.88	13.80±5.84	15.33±3.62
text-davinci-003	**90.50±0.78**	61.33±1.40	**66.34±2.45**	8.05±3.28	62.94±3.02	**66.29±0.76**	64.56±1.94
gpt-3.5-turbo	86.34±0.25	**63.38±0.67**	63.67±4.34	**28.60±2.12**	**67.32±0.92**	65.90±1.49	**66.60±1.21**
3-shot learning							
text-ada-001	3.88±1.87	9.47±2.98	12.57±5.00	8.19±4.62	13.25±1.56	7.58±3.93	8.73±2.88
text-babbage-001	10.20±6.50	2.08±2.08	10.04±8.14	12.00±1.00	16.09±1.79	7.08±4.25	9.29±4.58
text-curie-001	28.55±4.06	26.23±3.00	35.16±2.20	11.25±3.29	18.75±2.29	35.66±9.85	24.44±4.34
text-davinci-003	**90.88±1.19**	65.83±1.81	**74.24±0.90**	18.57±6.60	60.07±0.51	**73.16±1.69**	65.97±0.99
gpt-3.5-turbo	91.81±0.32	**69.09±1.66**	73.06±0.07	**43.98±0.45**	**74.06±0.86**	71.77±3.50	**72.83±1.39**

Fig. 4. F1 scores, Precision, and Recall for the first 20 articles from experiments for full article and per sentence prompt execution. For text-davinci-003 and gpt-3.5-turbo models provided results per zero, one, and three-shot sampling. For the spacy-RoBERTa model results are without shots.

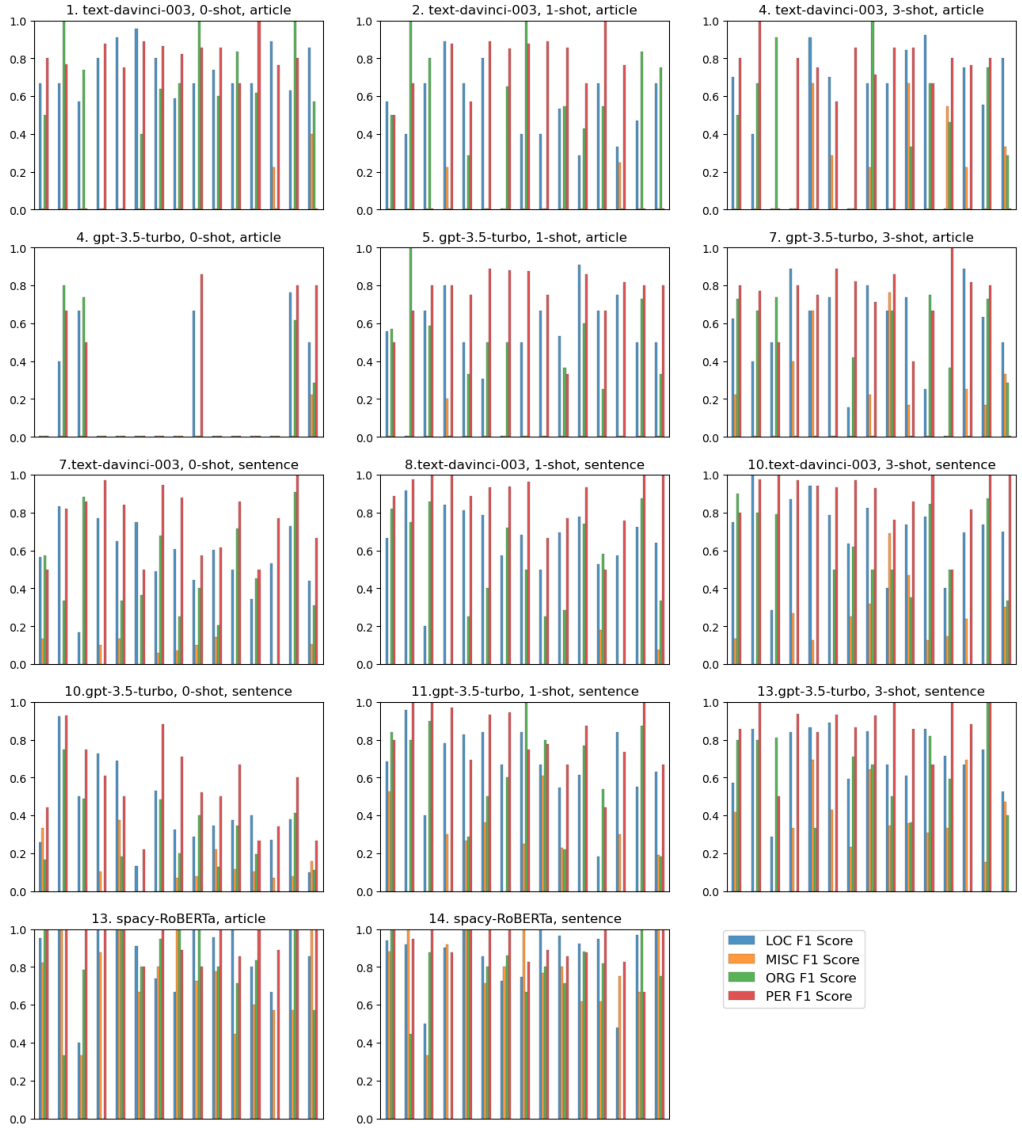

Fig. 5. F1 scores per each entity class for the first 15 articles from experiments for full article and per sentence prompt execution. For text-davinci-003 and gpt-3.5-turbo models provided results per zero, one, and three-shot sampling. For the spacy-RoBERTa model results are without shots.

Based on the results in Table 2 and Table 3, we can observe that text-ada-001, and text-babbage-001 showed the possibility to recognize named entities from the text in very rare cases. Overall, this is expected result for these models positioning as the most lightweight and

straightforward, capable to solve simple tasks. To increase their productivity, possible to experiment with decreasing request complexity by making prompts more straightforward and preparing samples with specific goals. Nevertheless, these models are from previous generations and, probably, soon will deprecate.

Regarding the results (Table 2-3, Fig. 4-5), the most recent and capable models, text-davinci-003 and gpt-3.5-turbo, show very close results on scenarios with sampling, but gpt-3.5-turbo is not capable to correctly recognize at all entities on full article prompts texts for most articles. Model text-davinci-003 shows close results with 0,1 and 3 sampling scenarios for both types of prompts: per document and per sentence. In fact, sampling had almost no impact on this model. Even more, 1-shot sampling has a worser result for document level scenario than 0-shot and 3-shot ones. Based on model characteristics, we could note, that the text-davinci-003 model has good context detection capabilities from the prompt request.

However, it is an unexpected finding, that the gpt-3.5-turbo model has similar performance to text-davinci-003 and even outperforms the last one in some cases. The key finding is that gpt-3.5-turbo model has great performance improvements with few-shot scenarios, while 0-shot scenario shows poor results. During experiments, random sampling was used, so experimenting with task-specific prompts could be used in the future.

One of the unexpected results is the poor quality of MISC class detection. In this case, detailed results analysis shows that this category is too abstract and could differ between different datasets and domains. By the way, with sampling models significantly increase recognition of the MISC named entities, especially gpt-3.5-turbo. In addition, PER class is the easiest to detect by GPT series models, while LOC and ORG have less performance. CoNLL dataset contains multiple content-dependent locations and organizations, and it could be challenging to detect this location or organization.

Table 4. lang-uk dataset. Scoring for experiments execution for sentence prompts. The biggest value in each block per shot is bolded.

algorithm	PER F1	ORG F1	LOC F1	MISC F1	Precision	Recall	F1
0-shot learning							
text-davinci-003	0.0	13.52	32.23	2.29	**8.6**	**50.84**	**14.71**
gpt-3.5-turbo	0.0	**16.25**	**34.34**	**3.26**	6.87	39.66	11.72
1-shot learning							
text-davinci-003	0.0	21.3	37.79	2.36	**12.53**	**50.84**	**20.11**
gpt-3.5-turbo	0.0	**19.58**	**58.54**	**6.0**	12.16	44.69	19.12
3-shot learning							
text-davinci-003	0.0	38.6	67.76	9.04	16.56	**56.42**	25.6
gpt-3.5-turbo	0.0	**40.0**	**68.57**	**16.51**	**26.42**	54.75	**35.64**

Lang-UK dataset. Table 4 contains **experiment** results for the lang-uk dataset with execution per sentence. We cannot run the experiment at the article level due to the prompt token count exceeding the limit (roughly 4000 tokens) in 2-4 times. Based on weak results for

text-ada-001, text-babbage-001, and text-curie-001 for CoNLL datasets, and the half of prompts exceed their limit (roughly 2000 tokens), we decide to not make tests with these models.

Based on the results of the experiments with Ukrainian texts (Table 4), we can observe a bit different result from CoNLL dataset experiments (Table 3). In contrast to CoNLL, which has more compact texts and many named entities, Ukrainian texts from the lang-uk dataset are more literature-like with rare entities. While precision and recall are close for CoNLL results, for the Ukrainian dataset we have very small precision compared to recall. Furthermore, we can see, that recall is comparable with results for CoNLL. It is an interesting finding, that for Ukrainian text we have significantly more detected entities, which do not exist in initial dataset labeling or are incorrectly detected. This is a good point for research in the future.

One of the unexpected findings for Ukrainian text is that gpt-3,5-turbo model outperforms text-davinci-003 model for the 3-shot learning approach. However, the error rate is too big for both models and it is too early to use these models for real-world tasks.

Another unexpected result was to have a 0.00 F1 score for PER investigation. A deeper look at results per prompt shows, that not all persons are labeled in the initial dataset. Hence, makes sense to prepare another dataset and run experiments on it.

Conclusion.

In this paper, named entity recognition by Open AI GPT series models, including GPT-3 (text-davinci-003) and GPT-3.5 (gpt-3.5-turbo), was investigated on two datasets: CoNLL 2003 for English text and lang-uk team dataset for Ukrainian. Overall, the obtained results show a high percentage of named entities recognition for unknown text. This demonstrates the significant potential of in-context learning and large language models possibilities to analyze text. Also, these models are sensitive to prompt request format, therefore prompt design is one of the keys to better recognition results. Moreover, including a few samples of the recognition leads to better results. However, GPT series models show significantly worse results compared to transformer-based models like RoBERTa, pre-trained for a specific dataset. This result is quite expected because some entity classes like MISC significantly depend on the specific dataset or domain context. Nevertheless, our expectations of similar performance in chosen Ukrainian dataset were not satisfied. Investigating this will be a critical area for future research.

At the same time, based on the recognition results of the named entities from an unknown text can find practical implementations in business. Solution using GPT-3 and GPT-3.5 can be implemented in short terms and give effort and avoid spending significant resources for training dataset preparation for other models.

To summarize further research directions, the most critical areas to investigate are how different structures of the prompt impact the results, experiments with various sampling approaches to select the right format, and samples pre-processing. Additionally, fine-tuning the models has been out of the scope of our research, but it could increase the performance of the models.

REFERENCES

[1] *Li Jing, Aixin Sun, Jianglei Han, and Chenliang Li*. "A survey on deep learning for named entity recognition." *IEEE Transactions on Knowledge and Data Engineering* 34, no. 1 (2020): 50-70.

[2] *Roy Arya*. "Recent trends in named entity recognition (NER)." *arXiv preprint arXiv:2101.11420* (2021).
[3] *Grishman Ralph, and Beth M. Sundheim*. "Message understanding conference-6: A brief history." In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. 1996.
[4] *LeCun Yann, Yoshua Bengio, and Geoffrey Hinton*. "Deep learning." *nature* 521, no. 7553 (2015): 436-444.
[5] *Yadav Vikas, and Steven Bethard*. "A survey on recent advances in named entity recognition from deep learning models." *arXiv preprint arXiv:1910.11470* (2019).
[6] *Shen Yanyao, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar.* "Deep active learning for named entity recognition." *arXiv preprint arXiv:1707.05928* (2017).
[7] *Vaswani Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin.* "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
[8] *Devlin Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.* "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
[9] *Baevski Alexei, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli*. "Cloze-driven pretraining of self-attention networks." *arXiv preprint arXiv:1903.07785* (2019).
[10] *Li Xiaoya, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li.* "Dice loss for data-imbalanced NLP tasks." *arXiv preprint arXiv:1911.02855* (2019).
[11] *Brown Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al.* "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
[12] *Wang Shuhe, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang.* "GPT-NER: Named Entity Recognition via Large Language Models." *arXiv preprint arXiv:2304.10428* (2023).
[13] *Ye Junjie, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui et al*. "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models." *arXiv preprint arXiv:2303.10420* (2023).
[14] *Sang, Erik F., and Fien De Meulder.* "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition." *arXiv preprint cs/0306050* (2003).
[15] Lang-uk team dataset for NER repository. URL: https://github.com/lang-uk/ner-uk (accessed on April 10, 2023)
[16] OpenAI homepage, access to UI prompt and API. URL: https://openai.com (accessed on April 25, 2023)
[17] Promptify library repository. URL: https://github.com/promptslab/Promptify (accessed on April 25, 2023)
[18] SpaCy, NLP framework homepage. URL: https://spacy.io/ (accessed on April 25, 2023)

ЕФЕКТИВНІСТЬ РОЗПІЗНАВАННЯ ІМЕНОВАНИХ СУТНОСТЕЙ ЗА ДОПОМОГОЮ КЛАСУ МОДЕЛЕЙ OPENAI GPT

Б. Павлишенко, І. Дроздов

Львівський національний університет імені Івана Франка,
вул. Драгоманова 50, 79005 Львів, Україна
bohdan.pavlyshenko@lnu.edu.ua, ihor.drozdov@lnu.edu.ua

Обсяг інформації дуже швидко зростає в усіх доступних джерелах, причому головною складовою в усьому обсягу інформації є текстові дані, тому обробка природної мови є однією з найбільш важливих галузей дослідження. Зростаючі обсяги інформації вимагають більш складних та ефективних моделей та підходів для ефективної обробки інформації. В той самий час, розпізнавання іменованих сутностей є однією з ключових складових в обробці тексту та відіграє важливу роль для розпізнавання тексту, автоматичної сумарізації тексту, перекладу та інших. На теперішній час є багато різних підходів до розпізнавання іменованих сутностей, однак запровадження так званої архітектури на основі трансформерів з механізмом уваги сприяло суттєвій зміні основних напрямків дослідження в сфері обробки природної мови, про що свідчить застосування трансформерів для досягнення найкращих результатів для більшості задач обробки природної мови. Тим часом, відносна простота, у порівнянні з іншими, архітектури тнасформерів дала можливість будувати великі мовні моделі з мільярдами параметрів, як, наприклад GPT-3.

Головна мета цієї статті – дослідити ефективність застосування декількох GPT моделей, створених компанією OpenAI, для розпізнавання іменованих сутностей в англомовному та україномовному текстах. Для дослідження використано один з найбільш популярних датасетів для такого типу досліджень CoNLL 2003 та датасет організації lang-uk, яка розмітила частину браунівського корпусу для задачі розпізнавання іменованих сутностей. Базуючись на відомих можливостях моделей GPT генерувати кращі результати у випадку наведених прикладів у вхідному запиті, експерименти були побудовані з використанням нуля, одного та трьох прикладів на кожен запит. Крім того, експеримети окремо проводилися як для всієї статті в одному запиті так і для кожного речення в цій статті окремими запитами для порівняння результатів за різним обсягом тексту в запиті. Для проведення експериментів, різні формати запитів були досліджені та один був обраний для всього експерименту. Оцінка результатів базується на F1 та специфіці результатів, які повертають моделі. Результати продемонстрували, в цілому, високу продуктивність найбільш нових моделей та збільшення продуктивності від старших до більш нових моделей. Більш того, результати демонструють, що є напрямки для подальшого покращення та дослідження.

Ключові слова: розпізнавання іменованих сутностей, обробка природної мови, GPT, OpenAI

Стаття надійшла до редакції: 12.06.2023
Прийнята до друку: 15.09.2023