UDC: 004.91

NAMED ENTITY RECOGNITION USING GENERATIVE TRANSFORMER MODELS WITH SENTENCE-LEVEL DATA AUGMENTATION APPROACHES

Ihor Drozdov*, Bohdan Pavlyshenko
Department of System Design,
Ivan Franko National University of Lviv
50 Drahomanova St., UA-79005 Lviv, Ukraine

Drozdov, I. V., Pavlyshenko, B.M. (2025). Named Entity Recognition Using Generative Transformer Models with Sentence-Level Data Augmentation Approaches. *Electronics and Information Technologies*, 31, 5–18. https://doi.org/10.30970/eli.31.1

ABSTRACT

Background. Named entity recognition, as one of the key tasks of the natural language processing (NLP) field, plays a vital role in the processing and understanding of the texts. Usage of transformer-based models demonstrates exceptional performance on most NLP tasks but requires a considerable amount of information for practical model training. Building a high-quality annotated dataset for named entity recognition is resource-intensive, especially for low-resourced languages. Using data augmentation to extend the annotated dataset with synthetic data provides an opportunity to increase the efficiency of the models for named entity recognition. This study aims to use sentence-level augmentations and large language models to improve model performance on small datasets.

Materials and Methods. To investigate the impact of data augmentation, 5%, 10%, and 20% of training data from the CoNLL and Ontonotes5 datasets with different characteristics were taken. Three main approaches were used to construct the augmented data: summarizing sentences using the T5 model, followed by inserting named entities, paraphrasing sentences using the OpenAI Api, and several methods of replacing named entities in initial and synthetic sentences. BERT, ALBERT, DistilBERT, and RoBERTa models were used for evaluation.

Results and Discussion. According to the results, the effectiveness of using different augmentation methods significantly depends on the initial dataset and its quality. For small datasets with few categories for recognition, sentence-level augmentation methods through summarization or paraphrasing improve the efficiency of models by up to 10%. On the other hand, with an increase in the size of the dataset, artificially created data can lead to a deterioration in recognition results.

Conclusion. Using data augmentation to recognize named entities is an effective tool for small datasets and can improve model performance in resource-constrained cases like specific domains and low-resourced languages. However, synthetic data cannot fully replace a larger, better-built original dataset through context extension for existing named entities and the generation of new, synthetic entities.

Keywords: named entity recognition, natural language processing, data augmentation, large language models

INTRODUCTION

Natural Language Processing (NLP) is a field of research that plays a crucial role in developing and improving existing information processing methods. As one of the fundamental tasks in NLP, Named Entity Recognition (NER) is essential for text understanding and extracting key information from textual data. Accurate recognition of entities like persons, organizations, locations, dates, and other categories is crucial for various NLP tasks like question answering, machine translation, sentiment analysis, and the construction of comprehensive knowledge bases [1, 2]. Recent advances in transformer-based architectures like BERT, RoBERTa, and others have substantially enhanced NLP tasks' performance in general and NER in particular by effectively capturing contextual and linguistic nuances within large-scale textual data [3, 4, 5]. Despite these advances, these models remain highly dependent on high-quality annotated datasets, and the lack of sufficient labeled data often impacts their effectiveness, especially in specialized domains and low-resourced languages.

Recent developments in generative Large Language Models (LLMs), such as GPT (OpenAI), LLaMA (Meta), and Mistral, provide significant advances for most of the NLP tasks. These models leverage unstructured textual data to extract and understand sophisticated language patterns. As a result, LLMs provide prompt-based communication in a native language, excellent text understanding capabilities, text generation, and advanced reasoning. Nevertheless, the adoption of LLMs is often limited by their computational resource requirements for request processing and model fine-tuning, very high associated infrastructure costs, and request processing speed, restricting their practical usage for many applications and smaller-scale deployments [6, 7]. Aside from this, LLMs' great reasoning possibilities greatly benefit smaller models' data preparation and fine-tuning.

One of the primary challenges in effectively training transformer-based models such as BERT or RoBERTa is the creation of sufficiently large datasets with high-quality annotated data. Typically, annotation is performed manually by data labeling specialists or domain experts. This process becomes even more complex when dealing with highly specialized domains or low-resource languages. Data augmentation offers a promising alternative for expanding labeled datasets by generating synthetic data [8, 9]. Data augmentation research has become increasingly popular in recent years and often requires selecting the most effective methods for specific tasks and application domains. Most augmentation techniques can be broadly categorized according to their application level: character level, word level, sentence level, and document level [10, 11]. Some methods are more general-purpose, while others are effective for specific classes of tasks. Nevertheless, selecting appropriate augmentation techniques for a given task can be challenging. In [12, 13], augmentation methods that have demonstrated exemplary performance for text classification tasks are reviewed; however, not all these methods are effective in the context of named entity recognition tasks.

The scope of this paper is to extend our previous research [14] from word-level data augmentations to sentence-level augmentations like text summarization, entity injection, and context-dependent LLM-based augmentations, generated by OpenAI models.

MATERIALS AND METHODS

In the scope of this research, CoNLL 2003 [15] and Ontonotes 5 [16] were used. Despite these datasets being introduced more than ten years ago, both are very popular for NER research as they provide an outstanding possibility to compare with other authors' research results. **Table 1** provides an overview of the CoNLL dataset, the number of sentences, and entities. It has four entity types: Person (PER), Location (LOC), Organization (ORG), and Miscellaneous (MISC). **Table 2** provides information about the Onotonotes5 dataset which has 18 different entity types: Person (PER), Facility, Nationalities/Religious/Political groups, GPE (Geo-political entity), Organizations (ORG), Locations (not GPE), Date (DATE), Time, etc. During this research, six subsets of the original training dataset were used, as demonstrated in **Table 3**.

Table 1. **Information about the CoNLL 2003 dataset.**

CoNLL 2003 dataset, English						
Set type	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	14,041	203,621	7,140	3,438	6,321	6,600
Validation set	3,250	51,362	1,837	922	1,341	1,842
Test set	3,453	46,435	1,668	702	1,661	1,617

Table 2. **Information about the Ontonotes5 dataset.**

Ontonotes5 dataset, English							
Set type	Sentences	Tokens	PER	ORG	GPE	DATE	Other*
Training set	59,924	1,088,442	6,292	5,363	1790	3,533	8,695
Validation set	8,528	147,718	1,071	914	506	799	1,641
Test set	8,262	152,723	977	950	462	767	1,750

Note: * – the “Other” column contains the total number of entities for the remaining 14 entity types.

Table 3. **Partial training subsets used in this work. Validation and test parts are unchanged.**

Abbre-viation	Sentences CoNLL	Sentences Ontonotes5	Description
S100	14,041	59,924	Contains all sentences from the original dataset without changes
S20	2,808	11,984	Contains 20% of the initial train dataset records: the first 10% of sentences and the last 10%
S10	1,404	5,992	Contains 10% of the initial train dataset records: the first 5% of sentences and the last 5%
S5	702	2,996	Contains only the first 5% of the dataset
R10	1,404	5,992	Contains 10% of the initial train dataset chosen by random
R5	702	2,996	Contains 5% of the initial train dataset chosen by random

Within the scope of this study, the main attention was paid to the creation of additional context through paraphrasing or summarization of sentences, the use of LLMs to create sentences close in context to the original ones, with the same entities from the original sentence. Additionally, two scenarios were used: replace named entities in the augmented sentence with random ones from the train dataset, and the second scenario – replace named entities in the augmented sentence with random ones generated by LLM.

For each data augmentation approach described above, the training sub-dataset term relates to the part of the dataset that was used during the training session. For example, dataset R5 (**Table 3**) has only 5% of randomly selected initial sentences from the original dataset and is a training sub-dataset for the R5 dataset from the original one. Augmentations were applied to sentences from the training sub-dataset with the multiplier provided in **Table 4**. Produced augmentations and original sentences were added to the new augmented training sub-dataset. Thus, the following data augmentation scenarios were used:

- Replace entities for random ones from the training sub-dataset (RND_ENT). With this augmentation type, the original positions and types of the named entities persisted. Still, each entity was replaced with a new, random one in the training sub-dataset.
- Rephrasing with OpenAI LLMs (OPENAI_SENT) – with this augmentation type, the OpenAI API was used to build the new sentences as a rephrasing of the original one using the same named entities, which exist in the original sentence. As a result, the new sentence extends the context for the original entities.
- Rephrasing or summarization with sentence entities (SUMM_SENT) – with this augmentation type, Text-to-Text summarization or rephrasing was used using the T5-base model [17] to build required sentences via sampling. All named entities from the original sentence were injected into random places for each newly generated sentence.
- Rephrasing with OpenAI LLMs with train sub-dataset entities (OPENAI_SDE) – this augmentation type was based on OPENAI_SENT augmentation. Random train sub-dataset entities were used to replace the existing ones in each.
- Rephrasing or summarization with train sub-dataset entities (SUMM_SDE) – this augmentation type is the same as OPENAI_SDE, but utilizes SUMM_SENT
- Rephrasing or summarization with OpenAI entities (SUMM_GENT) – this augmentation type was built the same way as SUMM_SDE but using OpenAI-generated entities.
- Rephrasing with OpenAI LLMs with OpenAI entities (OPENAI_GENT) – this augmentation type was built the same way as OPENAI_SDE but using OpenAI-generated entities.

Moreover, in [14] word-level augmentations like synonyms, antonyms, and word-embeddings were applied to the CoNLL dataset. During this experiment, the same data augmentation approaches on word-level were applied for the Ontonotes5 dataset to have the possibility to compare word-level augmentations with sentence-level ones for both datasets.

At the same time, our main objective is to investigate the impact of data augmentation on the most popular transformer-based models. Thus, BERT, RoBERTa, DistilBERT, and ALBERT models were used for fine-tuning during the experiment [14]:

- BERT model –one of the first transformer-based models, demonstrated state-of-the-art in multiple NLP tasks compared to previous approaches.
- ALBER and DistilBERT models aim to optimize the initial BERT model implementation with parameter optimization, faster training time, a smaller model, and similar performance due to more effective model utilization.
- RoBERTa model – also, a BERT-based model, optimizes the initial model, introduces additional dynamic masking, and uses a significantly bigger initial training dataset.

Experiment structure

Experiments were built and executed around the HuggingFace platform [18] as it provides valuable tools to load and manipulate datasets, prepare pipelines for dataset preparation, fine-tune process configuration, and model training and evaluation. **Fig.1** demonstrates a general scheme of the experiment flow to build the required datasets with augmentation and fine-tune models. The experiment has the following key points:

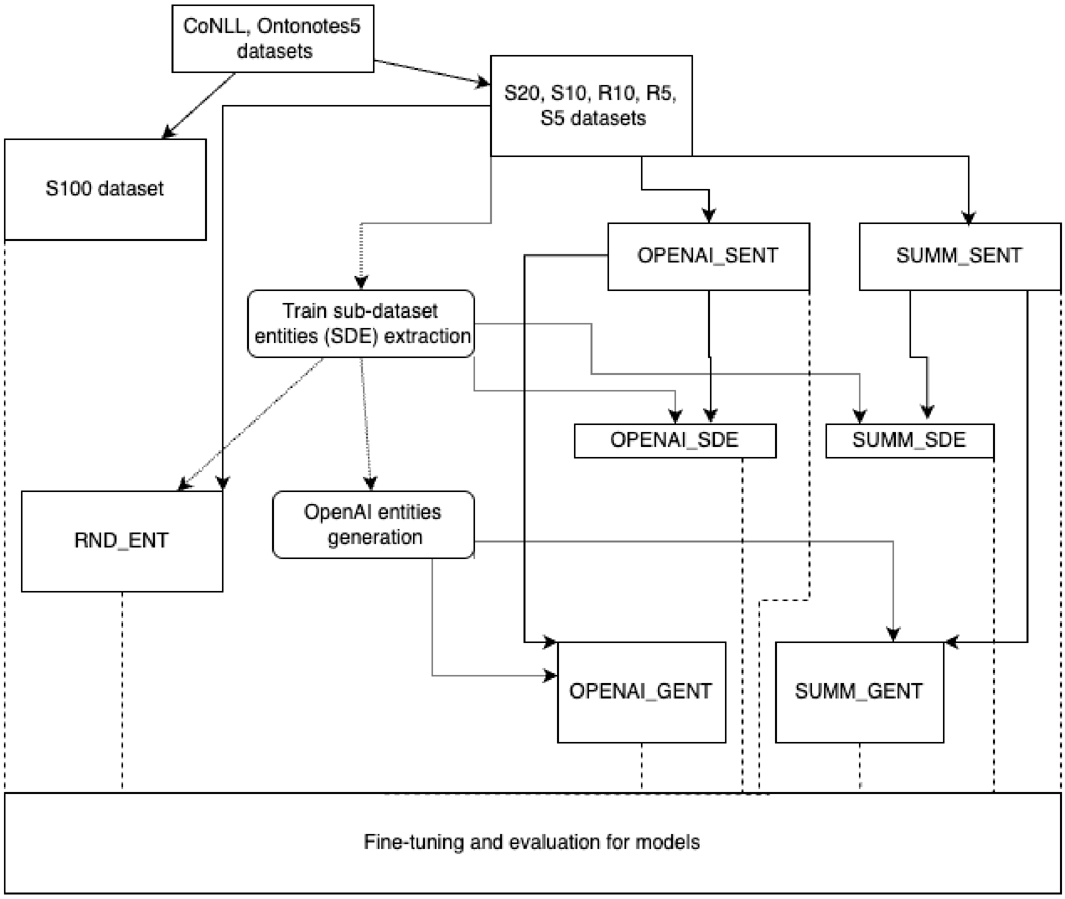
CoNLL, Ontonotes5 datasets
S20, S10, R10, R5, S5 datasets
S100 dataset
OPENAI_SENT
SUMM_SENT
Train sub-dataset entities (SDE) extraction
OPENAI_SDE
SUMM_SDE
RND_ENT
OpenAI entities generation
OPENAI_GENT
SUMM_GENT
Fine-tuning and evaluation for models

Fig. 1. Datasets generation and model fine-tuning scheme. Solid arrows – dataset was built from parent; Dotted arrows – relation to some additionally generated data to use; Dashed lines – flow direction to indicate which datasets were used for fine-tuning

- Dataset S100 was used only to receive the score of the models on the full dataset and compare it with the augmented ones.
- **Table 4** contains information about the number of additional augmented sentences per augmentation type, additionally generated for the augmented dataset. Original sentences were also included in the augmented dataset.
- The train sub-dataset entities (SDE) extraction process involves identifying all annotated named entities and constructing a dictionary for each type from the train sub-dataset.

Table 4. **Total augmentations count per augmented dataset, generated with different approaches.**

Scenarios	Augmentations count
RND_ENT	2, 5, 10
OPENAI_SENT, OPENAI_SDE, OPENAI_GENT	1, 3, 5
SUMM_SENT, SUMM_SDE, SUMM_GENT	3, 5

- OpenAI API [19] was used for all LLM prompt-based requests with the gpt-4.1-mini model.
- OpenAI's entity generation process uses entities from the train sub-dataset as a context and generates up to 4 additional entities per existing one with the same context.
- OPENAI_SENT and SUMM_SENT were built with additional sentences as described in **Table 4**. OPENAI_SDE, OPENAI_GENT, SUMM_SDE, and SUMM_GENT utilize augmented datasets from OPENAI_SENT and SUMM_SENT, but apply random entity replacements based on the augmentation type.
- The experiment used pre-trained models from the HuggingFace platform: bert-base-uncased, roberta-base, albert-base-v2, and distilbert-base-uncased.
- The following initial training parameters were used for fine-tuning models: batch size is 32, learning rate is $5e^{-5}$, weight decay is 0.01, and batch size is 16.
- For evaluation of the models F1 score was used with the seqeval library [20]
- All experiments were executed with Google Colab, a T4 GPU with High RAM runtime, or an Apple MacBook Pro M4 Max 40-core GPU.

RESULTS AND DISCUSSION

It makes sense to split the research results into two blocks:

- Datasets preparation – preparation of the datasets with different types of augmentations depends on available resources, the size of the initial dataset, and the time required to build the dataset. Let us shed light on some key points here.
- Model evaluation – It is essential to understand the behavior of the models during fine-tuning, evaluate the models' effectiveness based on the F1 score, and compare the results of different augmentation methods and initial sub-dataset sizes.

As expected, the preparation of augmented datasets differs significantly in terms of time and implementation complexity. **Table 5** demonstrates the required time to build 100 augmented sentences for the CoNLL dataset. The time needed to generate augmented entities was evaluated over 10 runs using 100 randomly selected sentences from the original dataset. It should be noted that this measurement is somewhat approximate, as various factors significantly influence it. Since RND_ENT utilizes only the training subset for inserting random entities, its execution is virtually instantaneous. The methods SUMM_SENT, SUMM_SDE, and SUMM_GENT demonstrate execution speeds more than 10 times faster than methods relying on the OpenAI API. It is important to note that the

Table 5. **Approximate time in seconds to create augmentations for 100 sentences for the CoNLL dataset.**

Augmentations count Augmentation type	2/5/10	1	3	5
RND_ENT	1	--	--	--
OPENAI_SENT, OPENAI_SDE, OPENAI_GENT	--	3,750	9,500	16,150
SUMM_SENT, SUMM_SDE, SUMM_GENT	--	--	750	1,150

summarization-based methods were executed using local computational resources. In contrast, the generation of augmentations via OpenAI depends entirely on the external platform's performance and response time.

Models Evaluation

In **Tables 6–9**, F1-score measurements for different models were demonstrated for all applied augmentation approaches and train sub-datasets. Additionally, for CoNLL and Ontonotes5 datasets, word-level augmentation results were added [14]. For each augmentation type, datasets were built with augmented sentence multipliers, as described in **Table 4**. In **Tables 6–9**, results were provided based on the mean and standard deviation for the augmentation type and dataset.

Table 6. **F1-scores for RoBERTa model evaluation and different CoNLL and Ontonotes5 datasets augmentation approaches.**

	S100	S20	S10	R10	S5	R5
CoNLL Dataset						
OPENAI_GENT		91.4±0.6	87.0±0.6	90.3±0.7	84.0±0.4	88.0±1.4
OPENAI_SDE		92.2±0.3	87.5±0.7	92.0±0.5	84.7±0.3	88.3±1.8
OPENAI_SENT		91.6±1.3	87.1±1.2	90.6±1.4	85.7±0.4	88.3±0.9
RND_ENT		92.8±0.2	87.9±0.2	92.5±0.5	86.0±0.3	90.0±2.1
SUMM_GENT		90.1±0.2	83.0±0.0	87.5±0.0	77.8±0.6	84.3±1.9
SUMM_SDE		90.2±1.0	82.3±0.3	88.6±1.1	76.4±1.4	82.6±3.3
SUMM_SENT		91.2±0.3	85.7±0.4	89.5±0.1	81.6±1.5	86.5±2.3
Word-level		92.7±0.3	88.6±0.3	92.0±0.4	86.1±0.8	89.4±0.5
Without aug.	95.77	91.61	87.46	90.22	82.92	83.16
Ontonotes5 Dataset						
OPENAI_GENT		80.3±1.1	77.7±1.2	82.0±1.4	77.5±1.1	79.9±1.1
OPENAI_SDE		80.4±1.2	78.9±1.3	82.9±0.8	79.6±0.9	82.0±0.4
OPENAI_SENT		78.0±3.3	76.2±2.6	80.2±3.6	76.7±2.1	77.7±3.2
RND_ENT		81.1±1.3	79.7±1.0	83.5±0.4	80.4±0.1	82.6±0.1
SUMM_GENT		81.0±0.3	79.0±0.4	83.5±0.4	78.2±0.1	80.0±0.0
SUMM_SDE		80.1±0.1	78.1±0.1	83.5±0.6	78.3±0.6	81.1±0.2
SUMM_SENT		82.6±0.1	80.6±0.2	84.1±0.6	80.1±0.6	81.5±0.8
Word-level		82.7±0.2	81.0±0.4	85.3±0.5	80.9±0.4	83.2±0.2
Without aug.	88.62	83.5	81.4	84.71	79.83	82.67

Table 7. **F1-scores for BERT model evaluation and different augmentation approaches for CoNLL and Ontonotes5 datasets.**

	S100	S20	S10	R10	S5	R5
CoNLL Dataset						
OPENAI_GENT		89.4±0.9	83.3±1.1	87.7±0.6	80.8±0.4	86.3±1.5
OPENAI_SDE		89.9±0.4	84.3±0.8	89.5±0.6	80.4±0.9	86.8±1.5
OPENAI_SENT		89.4±1.9	85.0±2.1	89.0±1.5	81.0±0.3	86.0±1.0
SUMM_GENT		87.5±0.5	80.8±0.0	84.2±1.2	73.8±0.1	80.6±1.2
SUMM_SDE		86.9±0.2	80.1±0.7	85.3±0.8	71.3±1.8	79.9±0.6
SUMM_SENT		88.0±0.3	81.2±0.7	86.0±0.6	74.4±0.1	81.3±1.2
RND_ENT		90.6±0.2	84.2±0.5	90.2±0.4	81.5±0.7	87.0±0.7
Word-level		90.7±0.3	85.2±0.7	90.0±0.2	82.3±1.0	87.5±0.7
Without aug.	94.4	90	85.5	88.9	77.2	77.6
Ontonotes5 Dataset						
OPENAI_GENT		75.8±1.2	73.3±1.4	77.3±1.8	73.6±0.8	75.1±0.8
OPENAI_SDE		76.1±1.8	74.1±1.3	78.9±1.1	74.8±0.4	77.3±0.7
OPENAI_SENT		72.6±3.8	71.1±2.9	75.9±3.8	71.8±2.5	73.1±2.6
SUMM_GENT		76.7±0.1	74.0±0.0	77.8±0.3	72.9±0.3	74.9±0.3
SUMM_SDE		74.6±0.6	72.7±0.1	79.1±0.7	73.4±0.1	75.9±0.0
SUMM_SENT		76.6±0.3	75.1±0.5	79.3±1.0	74.2±0.0	76.6±0.5
RND_ENT		77.2±0.6	76.1±0.8	79.7±0.4	76.6±0.1	78.3±0.6
Word-level		78.3±0.4	76.1±0.8	81.5±0.3	75.9±0.7	79.0±0.4
Without aug.	85.4	80	76.8	81.4	75	76.8

Table 8. **F1-scores for ALBERT model evaluation and different augmentation approaches for CoNLL and Ontonotes5 datasets.**

	S100	S20	S10	R10	S5	R5
CoNLL Dataset						
OPENAI_GENT		87.0±1.2	82.7±0.5	86.3±1.4	80.8±0.4	83.8±0.8
OPENAI_SDE		88.0±1.1	82.5±0.4	88.4±0.2	80.9±1.0	84.6±1.6
OPENAI_SENT		88.2±2.1	83.6±2.6	87.2±1.6	81.1±1.8	83.5±0.9
SUMM_GENT		86.3±0.4	78.7±1.0	82.7±0.6	76.2±0.2	79.6±0.2
SUMM_SDE		85.2±0.3	78.6±0.1	83.7±0.1	74.0±0.1	80.0±1.0
SUMM_SENT		87.2±0.9	82.2±0.1	85.6±0.3	77.6±0.1	81.5±0.2

	S100	S20	S10	R10	S5	R5
RND_ENT		88.1±0.4	83.7±1.4	88.4±0.1	81.1±0.6	84.6±0.2
Word-level		88.6±0.7	83.3±0.6	88.6±0.7	82.6±0.6	85.1±0.9
Without aug.	93.4	89.7	85.7	87.5	81.9	83.9
Ontonotes5 Dataset						
OPENAI_GENT		72.8±2.0	71.5±1.7	74.8±1.6	71.6±2.5	73.0±1.0
OPENAI_SDE		73.4±2.2	71.5±2.0	76.5±1.2	72.8±1.9	75.4±0.8
OPENAI_SENT		71.9±3.8	70.4±2.8	73.7±3.8	70.0±3.1	71.7±3.1
SUMM_GENT		75.1±0.6	73.0±0.3	76.8±0.2	68.5±5.4	73.9±0.6
SUMM_SDE		73.8±0.2	71.5±0.9	77.9±0.7	70.9±1.0	74.2±0.7
SUMM_SENT		76.4±0.4	74.7±0.1	78.8±0.2	74.3±0.1	75.8±0.1
RND_ENT		74.4±1.8	72.0±2.3	77.7±1.3	73.5±1.9	76.8±1.0
Word-level		75.2±0.5	72.9±0.8	79.2±0.3	73.8±0.5	76.7±1.2
Without aug.	84.5	78.2	76.1	80	75.5	76.5

Table 9. **F1-scores for DistilBERT model evaluation and different augmentation approaches for CoNLL and Ontonotes5 datasets.**

	S100	S20	S10	R10	S5	R5
CoNLL Dataset						
OPENAI_GENT		88.2±0.2	81.8±0.3	86.2±0.3	78.2±2.7	83.7±1.9
OPENAI_SDE		89.2±0.1	83.4±0.2	88.3±0.8	78.5±2.6	84.3±1.6
OPENAI_SENT		88.8±1.2	83.1±1.7	87.8±0.7	79.9±1.7	83.3±1.3
SUMM_GENT		85.2±0.7	76.1±0.9	80.7±0.7	70.1±1.7	75.2±2.6
SUMM_SDE		84.5±0.7	76.7±0.3	81.8±0.0	68.3±1.2	75.5±2.0
SUMM_SENT		85.5±0.2	78.2±0.6	83.6±0.7	73.1±0.4	78.3±0.7
RND_ENT		89.7±0.4	84.0±0.4	89.0±0.5	81.5±0.7	85.0±1.3
Word-level		89.7±0.3	84.2±0.6	89.1±0.3	81.4±0.7	85.9±0.3
Without aug.	94.2	89.9	83.9	87.7	73.3	77.7
Ontonotes5 Dataset						
OPENAI_GENT		74.1±1.3	71.7±1.6	75.9±1.6	71.1±0.2	72.9±0.6
OPENAI_SDE		74.6±1.5	72.3±1.2	77.7±0.6	72.7±0.3	75.6±0.5
OPENAI_SENT		71.9±3.8	70.3±2.3	74.8±3.2	70.4±1.9	72.0±2.2
SUMM_GENT		74.6±0.6	71.0±0.5	75.6±0.1	69.8±0.3	71.2±0.5

	S100	S20	S10	R10	S5	R5
SUMM_SDE		72.9±0.7	69.7±1.3	76.9±0.4	70.0±0.5	72.9±0.0
SUMM_SENT		76.4±0.4	73.0±0.1	77.9±0.3	72.1±0.3	74.9±0.4
RND_ENT		75.8±0.6	74.1±1.2	78.7±0.7	74.4±0.2	76.9±0.3
Word-level		77.3±0.6	75.2±0.6	80.4±0.3	74.4±0.5	77.6±0.2
Without aug.	85.3	78.4	76	79.6	74.1	75.3

Based on **Tables 6–9**, the following key points can be highlighted:

- The performance of the models on the S100 dataset exceeds the best result of all sub-datasets, even with augmentations, by 3–4%. This is an expected outcome, as the full dataset contains more diverse named entities in various contexts.
- For S20, S10, R10 patterns, results for the CoNLL dataset demonstrated a positive impact for some augmentation methods and a negative impact for others. In most cases, word-level augmentations and augmentations based on random entity replacement show minor improvement in 1-3% for RoBERTa and BERT models. At the same time, ALBERT and DistilBERT even have a negative impact. All models, except RoBERTa, show an adverse effect of the augmentation on dataset quality and the received results.
- For the Ontonotes5 dataset, all models demonstrated a negative impact on the performance of the models. The Ontonotes5 dataset is bigger than CoNLL, has more distributed entities by dataset, and has 18 categories. Augmented data with the same entities or injected out-of-context entities improves some popular categories but negatively impacts the rest.
- For the S5 pattern on the CoNLL dataset, most methods demonstrated performance improvement up to 5%. It is not the expected result that augmentation methods, based on summarization, demonstrated negative impact and performance degradation up to 5-7% in some cases. From the group of those methods, summarization with random entity injection demonstrates better results with negligible performance degradation. Also, the ALBERT model hurts this dataset pattern in all cases, except word-level.
- For the R5 pattern on the CoNLL dataset, most of the augmentations demonstrated a significant increase in performance up to 7-8% in almost all cases, except the summarization group of methods. The summarization group of methods demonstrated minor improvement for RoBERTa and BERT models and performance degradation for the rest of the models. Some of the augmentation methods achieved performance results on the level of S10, R10, S20 datasets, or close to it.
- It is interesting that the context in which the named entity is used has a bigger impact on the model's possibilities to recognize some entity than the named entity itself. Sentences rephrasing using the same entities in the original sentence with OpenAI demonstrated better results in almost all experiments than summarization methods with entity injection. Moreover, for both approaches, entity replacement without random ones from the dataset or OpenAI-generated ones doesn’t make any performance improvements and has an adverse effect.
- It is not an expected finding that summarization and OpenAI-based methods do not provide performance improvement on entity recognition compared with simpler ones, like word-level.

CONCLUSION

This article investigates the impact on the effectiveness of sentence-level data augmentation models, the generation of augmented data using LLMs using OpenAI models, and the substitution of entities in sentences with random ones from a set of entities. To investigate how the size of the initial dataset affects the efficiency of augmentation, the CoNLL 2003 and Ontonotes5 datasets were used, and six different approaches of the initial dataset splittings were used for each of them: a complete dataset for reference comparison, 20% of the initial training dataset, two split approaches by 10% and two 5% partitions. For the 10% and 5% datasets, two different methods were used: all sentences from the initial dataset in a row, as this allows taking related sentences into one dataset, and sentences taken randomly. The study used the RoBERTa, DistilBERT, ALBERT, and BERT models, with the estimate based on the F1 score.

Creating additional context for named entities using summarization or paraphrasing techniques of the initial sentences, while preserving the initial named entities, should increase the efficiency of the models. According to the results obtained, in most cases, word-level augmentations showed better results or were on a par with sentence-level methods. It is worth noting that augmentations based on the substitution of entities in the initial sentences showed results similar to methods at the word level. However, in most cases, sentence-level methods negatively impacted model performance more than a reference result without augmentations for a comparable data set. A significant increase in the context of applying specific named entities leads to an oversaturation of the initial dataset with certain entities but reduces the model's generalization ability.

Most methods showed a significant improvement in results for the CoNLL dataset when 5% of the initial dataset was split. Almost all augmentation methods showed an increase of 5-9%, for the RoBERTa model using augmentations based on random entity substitution, 92.26% was achieved compared to a non-augmentation option of 83.16%. Nevertheless, the results of using augmentations for the Ontonotes5 dataset did not improve the result, and in many cases, worsened it. Due to the larger size of the initial dataset, the relatively small number of named entities in the text, and the large number of categories, augmentation is inefficient. It leads to a decrease in data quality for the Ontonotes5 dataset.

To summarize further research directions, context generation looks promising to create synthetic data, but requires more careful planning. Named entities distribution and initial dataset quality could significantly impact the result. Using LLMs as a supporting tool for training dataset generation for smaller models could provide excellent results, especially with fine-tuning models like LLaMa and Mistral for domain analysis, existing dataset analysis, and extending the weakest parts.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, authorship, and/or publication of this article.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [I.D.]; methodology, [I.D.]; validation, [B.P., I.D.]; formal analysis, [I.D.].; investigation, [I.D.]; resources, [I.D.]; data curation, [I.D.]; writing – original draft preparation, [I.D.]; writing – review and editing, [B.P.]; visualization, [I.D.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of NAACL-HLT 2016*, 260–270. https://doi.org/10.18653/v1/N16-1030

[2] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[4] Yamada, I., Asai, A., Shindo, H., Takeda, H., & Matsumoto, Y. (2020). LUKE: Deep contextualized entity representations with entity-aware self-attention. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 6442–6454. https://doi.org/10.18653/v1/2020.emnlp-main.523

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*. https://doi.org/10.48550/arXiv.1907.11692

[6] Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems,* 33 (2020): 1877-1901.

[7] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. https://doi.org/10.48550/arXiv.2302.13971

[8] Chen, J., Tam, D., Raffel, C., Bansal, M., & Yang, D. (2023). An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics*, *11*, 191-211. https://doi.org/10.1162/tacl_a_00542

[9] Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 968–988, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.84

[10] Chen, S., Aguilar, G., Neves, L., & Solorio, T. (2021). Data augmentation for cross-domain named entity recognition. *arXiv preprint arXiv:2109.01758*. https://arxiv.org/abs/2109.01758

[11] Dai, X., & Adel, H. (2020). An Analysis of Simple Data Augmentation for Named Entity Recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3861–3867. International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.343

[12] Pavlyshenko, B., & Stasiuk, M. (2023). Augmentation in a binary text classification task. In *2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT)* (pp. 177–180). IEEE. https://doi.org/10.1109/ELIT57602.2023.10151742

[13] Pavlyshenko, B., & Stasiuk, M. (2024). Data augmentation in text classification with multiple categories. *Electronics and Information Technologies*, 25, 67–80. http://dx.doi.org/10.30970/eli.25.6

[14] Pavlyshenko, B., & Drozdov, I. (2024). Influence of data augmentation on named entity recognition using transformer-based models. *Electronics and Information Technologies*, 28, 61–72. http://dx.doi.org/10.30970/eli.28.6

[15] Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*. https://arxiv.org/abs/cs/0306050

[16] Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers* (pp. 57–60). https://aclanthology.org/N06-2015

[17] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*(140), 1–67. http://jmlr.org/papers/v21/20-074.html

[18] HuggingFace. (n.d.). HuggingFace [Computer software]. Retrieved June 2025, from https://huggingface.co/

[19] OpenAI. (n.d.). OpenAI API [Computer software]. Retrieved June 2025, from https://platform.openai.com/docs/api-reference

[20] Seqeval library repository. (n.d.). Retrieved June 2025, from https://github.com/chakki-works/seqeval

РОЗПІЗНАВАННЯ ІМЕНОВАНИХ СУТНОСТЕЙ ІЗ ВИКОРИСТАННЯМ ГЕНЕРАТИВНИХ ТРАНСФОРМЕРНИХ МОДЕЛЕЙ З АУГМЕНТАЦІЄЮ ДАНИХ НА РІВНІ РЕЧЕНЬ

Ігор Дроздов, Богдан Павлишенко
Кафедра системного проектування,
Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, м. Львів, 79005, Україна

АНОТАЦІЯ

Вступ. Розпізнавання іменованих сутностей як одне з ключових завдань галузі обробки природної мови відіграє важливу роль в опрацюванні та розумінні текстів. Використання моделей на основі трансформерів демонструє виняткову продуктивність у більшості задач в області опрацювання природної мови, але вимагає значного обсягу інформації для практичного навчання моделі. Створення високоякісного анотованого набору даних для розпізнавання іменованих сутностей потребує значних ресурсів, особливо для малопоширених мов. Використання аугментації даних для розширення анотованого набору даних синтетичними дає змогу підвищити ефективність моделей для розпізнавання іменованих сутностей. Метою цього дослідження є використання аугментації на рівні речень та великих мовних моделей для підвищення продуктивності моделей на малих наборах даних.

Матеріали та методи. Для дослідження впливу аугментації даних було взято 5%, 10% і 20% відсотків тренувальних даних із датасетів CoNLL та Ontonotes5 з різними характеристиками. Для побудови наборів аугментованих даних було використано три основні підходи: узагальнення речень за допомогою моделі T5 із подальшою вставкою іменованих сутностей, перефразування речень за допомогою OpenAI Api, а також кілька методів заміни іменованих сутностей у початкових і синтетичних реченнях. Для оцінювання використовували моделі BERT, ALBERT, DistilBERT і RoBERTa.

Результати. Ефективність використання різних методів аугментації суттєво залежить від початкового набору даних і його якості. Для невеликих наборів даних із невеликою кількістю категорій для розпізнавання використання методі аугментації на рівні речень за допомогою узагальнення або перефразування дають підвищення ефективності моделей до 10%. З іншого боку, у випадку штучно створених даних збільшення обсягу набору даних може призвести до погіршення результатів розпізнавання .

Висновки. Використання аугментації даних для розпізнавання іменованих сутностей є ефективним інструментом для невеликих наборів даних і може підвищити продуктивність моделі у випадках з обмеженими ресурсами. Однак синтетичні дані не можуть повністю замінити більший, краще побудований вихідний набір даних за допомогою розширення контексту для існуючих іменованих сутностей і генерування нових синтетичних сутностей.

Ключові слова: розпізнавання іменованих сутностей, обробка природної мови, аугментація даних, великі мовні моделі

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
25 June 2025	10 July 2025	01 August 2025	31 October, 2025