Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6223–6231
Marseille, 20-25 June 2022

Named Entity Recognition to Detect Criminal Texts on the Web

Paweł Skórzewski, Mikołaj Pieniowski, Grażyna Demenko,
Adam Mickiewicz University, Poznań
ul. Wieniawskiego 1, 61-712 Poznań
pawel.skorzewski@amu.edu.pl, mikpie1@st.amu.edu.pl, lin@amu.edu.pl,

Abstract

This paper presents a toolkit that applies named-entity extraction techniques to identify information related to criminal activity in texts from the Polish Internet. The methodological and technical assumptions were established following the requirements of our application users from the Border Guard. Due to the specificity of the users' needs and the specificity of web texts, we used original methodologies related to the search for desired texts, the creation of domain lexicons, the annotation of the collected text resources, and the combination of rule-based and machine-learning techniques for extracting the information desired by the user. The performance of our tools has been evaluated on 6240 manually annotated text fragments collected from Internet sources. Evaluation results and user feedback show that our approach is feasible and has potential value for real-life applications in the daily work of border guards. Lexical lookup combined with hand-crafted rules and regular expressions, supported by text statistics, can make a decent specialized entity recognition system in the absence of large data sets required for training a good neural network.

Keywords: criminal texts, named entity recognition, natural language processing

1. Introduction

In recent years, the problem of combating organized crime related to cross-border criminal activities (e.g., illegal smuggling of various goods across borders, particularly drugs, and other criminal activities like slavery, prostitution, trafficking in human organs) has become particularly important. Valuable criminal-justice analyses based on web texts are currently difficult to be automatically accessed and used by intelligence investigators. In their daily work, employees of the Border Guard manually collect, read, and analyze dozens of documents from various websites. Most of these documents are not relevant to the desired activity, e.g., drug trafficking. The process is highly time-consuming, and the results are not necessarily satisfactory. It is necessary to automate searching for information, extracting the desired information, classifying documents, and profiling the text authors and organized groups. Informal text coming from heterogeneous sources is hard to process entirely automatically. Therefore, identifying and recognizing entities, i.e., places, organizations, or personal names, could help police and Border Guard officers understand and find relevant information in the data extracted.

The problem of recognizing named entities is well researched (Minkov, Wang, and Cohen, 2005), also in terms of language-independent solutions. Even so, only preliminary work has been presented for crime detection based on web text.

An excellent overview of the literature dedicated to NER extraction is given by Al-Moslmi et al. (2020). The paper provides an overview of state of the art in this area, including Named Entity Recognition (NER), Named Entity Disambiguation (NED), and Named Entity Linking (NEL). The authors not only explain the concept of NER in detail but also introduce the concept of NED, which refers to the ambiguity of the extracted units, and NEL, which refers to the mutual relations of the identified units.

Web texts are particularly useful in domain-specific applications because they contain information that may not be available in well-structured databases. However, such information is frequently hidden in unstructured text, thus limiting its usage in criminal activity detection. Despite the availability of generic named entity recognition tools, analyzing short informal texts collected from networks signaling illegal activities of individuals or organized crime groups poses the following challenges (Chau, Xu, and Chen, 2002).

- The analysis of texts in terms of crime-related content requires recognizing not only standard named entity categories like person names, organizations, and locations. Other expression categories, such as addresses, product names (e.g., drug names, cigarette and alcohol brands), descriptions, or actions, are equally relevant to crime intelligence analysis. Therefore, it is necessary to define the taxonomy of named entities precisely.
- Web texts are very specific: they are very noisy compared to other types of text data, such as well-prepared documents. They are mostly very short, contain many typos, spelling errors, and different kinds of grammatical errors, thus making the entity extraction task very difficult.
- The vocabulary is full of domain jargon and numerous ambiguities at every level of linguistic analysis.

These conditions are more or less important, depending on the language. For the Polish language, irregular grammar, numerous ambiguities at each linguistic level, and a relatively free sentence order sometimes make it impossible to interpret the extracted information correctly, even in the case of human analysis. Methods used in named entity recognition systems for Polish include conditional random fields (Waszczuk et al., 2013; Marcińczuk, Kocoń, and Oleksy, 2017) and recurrent neural networks (Borchmann, Gretkowski, and Graliński, 2018; Marcińczuk, Kocoń, and Gawor, 2018).

We discuss the following research questions based on the analysis of actual crime text collected from the Internet. Section 2 presents the applied methodology, including the list of entity categories we distinguished and the named entity extraction approaches we used. In Section 3, we discuss the process of gathering and annotating our experimental dataset and present their results. Section 4 describes the algorithms and models we developed to extract named entities from the text. In Section 5, we present the evaluation results. Section 6 contains the discussion of the obtained results and the further plans.

2. Methodology

2.1 Entity Categorization

Named entities considered in the context of analyzing texts in terms of crime-related expressions are not limited to proper names but also include descriptions, names of actions, etc. We have distinguished nine categories pertinent to the task:

- **Identifier** is a named entity allowing to directly identify the author of the text, e.g., first name, last name, nickname, telephone number. This category also includes all named entities that facilitate locating the event or identifying the author of the text, e.g., e-mail addresses, links, URLs, website names, etc. Examples: *Zbyszek* (a given name), *zb@example.com* (an e-mail address), *+48123456789* (a phone number).
- **Object–person**. This category includes all terms referring to people, such as demonyms, ethnonyms, names of professions, functions, nationalities, etc. Examples: *Polak* ('Pole'), *żoliborzanin* ('inhabitant of Żoliborz district'), *starzec* ('old man'), *sekretarka* ('secretary').
- **Object–thing**. This category is related to potential crime objects, excluding people. It includes physical or virtual (e.g., data) goods, works, and products – anything that may be the object of trafficking or a crime. A variety of entity types belongs to this category: names of drugs and medical substances, alcohols, guns, documents, vehicle brands, etc. Examples: *dowód osobisty* ('identity card'), *LSD*, *tequila*, *Volkswagen*
- **Action**. This category includes the names of actions directly or indirectly related to criminal activity. Actions directly associated with illegal activity include selected types of crimes, e.g., smuggling or drug trafficking. Activities indirectly related to criminal activity include, for example, traveling, accessing information, using websites, or shipping goods. Actions can be expressed with verbs (e.g., *wysłać* 'to send') or nouns (e.g., *wysyłka* 'shipment'). Verbs denoting actions refer to dynamic situations, i.e., situations that involve a change in the state of the performer of this activity, the object to which the activity relates, or the relationship between the participants of the action. These actions are carried out consciously, i.e., under the contractor's control. They include verbs for movement (*walk*, *drive*, *move*, *carry*), making sounds (*talk*, *whisper*, *cry*), judging (*praise*, *condemn*), physical activity (*work*, *beat*, *pull*), and much more. Actions are generally expressed by verbs denoting activities or dynamic situations. Verbs in the first person are particularly valuable, e.g., *sprzedam* ('I will sell'), *kupiłem* ('I bought'). Actions can also be signaled indirectly with other parts of speech, e.g., *sprzedaż i kupno* ('sale and purchase'), *wymiana* ('exchange'), *handel* ('trade'), *dystrybucja* ('distribution').
- **Organization**. This category includes the names of major Polish and international organizations and organizations related to cross-border smuggling. Examples: *Straż Graniczna* ('Border Guard'), *WORD* ('Voivodship's Road Traffic Center').
- **Location** category includes geographical places, addresses, and names of institutions. Examples: *Warszawa* ('Warsaw'), *ul. Słowackiego 8* (an address).
- **Time**. This category includes temporal expressions of various kinds, such as date (an expression that describes the appointment according to the calendar), time (exact hour/minutes), time of day/night (does not have to be very precise), duration (a time interval that answers the question "how long"), or set (an expression that describes a series of events; it answers the question "how often").
- **Measure**. This category includes terms relating to size: physical measures, terms indicating the size, numbers concerning specific items, also names of currencies. Measures can be expressed with different parts of speech. Examples: *100 dolarów* ('100 dollars'), *5 zł* ('5 zlotys'), *200 mg*, *5 szt.* ('5 pieces')
- **Description** category includes various expressions of characteristics, explanation, comments and can be expressed with different parts of speech. Examples: *bezpośrednio od producenta* ('directly from the manufacturer'), *białe* ('white'), *z zagranicy* ('from abroad'), *tanie w dobrej cenie* ('cheap at a good price'), *bez akcyzy* ('duty free').

2.2 Named Entity Extraction Approaches

We can distinguish three primary named-entity extraction approaches: based on lexicon lookup and rules, statistical approach, and machine learning. The last concept includes neural networks, which can automatically infer features through deep learning.

Rule-based systems rely on hand-crafted rules that do not require annotated training data since they depend on lexical resources. These rules can be structural, contextual, or lexical (Krupka and Hausman, 1998). Their precision can become high because of the lexicons and domain-specific knowledge. The disadvantage is that this also makes them domain-dependent, that lexicon resources may be unavailable, and that constructing and maintaining such resources for many languages is costly.

Statistic-based systems use statistical models to identify specific patterns or cues for entities in texts and require a training data set to obtain the statistics. Such systems may use a statistical language model to identify named entities in texts (Witten et al., 1999).

Machine-learning-based systems rely on entropy maximization (Borthwick et al., 1998), neural networks (Lample et al., 2016), decision trees (Baluja, Mittal, and Sukthankar, 2000), hidden Markov models (Miller, Leek, and Schwartz, 1998), or other machine learning techniques. Deep learning methods can be used to infer features automatically. Neural networks do not need seeds, ontologies, or domain-specific lexicons and are therefore more domain-independent. However, building robust models require large datasets.

Instead of relying on a single approach, our named entity extraction system utilizes a combination of lexicon lookup, hand-crafted rules, statistics, and neural networks.

3. Experimental Data

The entire process of preparing the evaluation corpus can be divided into three key steps. The steps include locating the sources of adequate texts, extracting and storing the found data, and finally, annotating gathered texts. The following subsections describe the process in an analogous order.

3.1 Text Collecting

To create an evaluation corpus of the Named Entity Recognition algorithm, it was necessary to collect texts rich in domain lexis corresponding to the criminal environment and containing as many of the categories listed in section 2.1 as possible. Depending on the source of given texts, either a manual or automatic approach was employed to capture them.

3.1.1 Indication of potential text sources

The first step towards creating an evaluation corpus was to identify potential sources of such texts. The sources used in the process can be divided into two main categories - those originating from the Clearnet and those originating from the anonymous TOR (The Onion Router) network. By making it impossible to trace a user's movements, the TOR network allows them to use the web anonymously[1]. This feature allows the network to be used by criminals to provide or use illegal services. To access content published on the TOR network, it is necessary to install a suitable browser that allows such access. TOR sites with illegal content are usually secured and require registration (Mider, 2019). Moreover, the URLs of TOR networks are usually strings of random characters - letters and numbers ending with the domain ".onion" (Krauz, 2017), which was an additional complication in the text collection process. These factors meant that the texts found in the pre-imposed requirements were collected manually rather than automatically, as was the case with those from Clearnet. In this case, the manual method was much more effective than developing a script that bypassed the safeguards mentioned above.

In order to determine Polish sources of criminal texts meeting the established criteria, a list of active Polish sites operating in the TOR network, published by the ITcontent[2] service, was used. The Polish sites Cebulka and Darknet were selected from this list. However, international sites such as Apollon Market or Dream Market were also used as text sources.

In the case of Clearnet, the traditional Google search engine was used to find texts. Mainly two-part queries consisting of an action verb and a following illegal object were used. Of the results returned, the most relevant were selected. Their content was then searched analogously to locate more texts. The following list presents the sources used for linguistic data extraction.

- **Cebulka** – the most popular Polish discussion board and auction site operating in the TOR network. Partial access is available without registration.
- **Darknet (Polka)** – Polish discussion board and auction site operating in the TOR network. Access to the content requires registration.
- **Apollon Market / Dream Market / White House Market –** international auction services operating in the TOR network. Each of them requires registration for the contents to be viewed.
- **oglaszamy24h.pl / top-ogloszenia.pl –** Polish advertising websites operating in the Clearnet. Registration is not required.
- **dopalacze-sklep.org –** Polish online shop offering illegal drugs of various kinds. It operates in the Clearnet and does not require registration.

3.1.2 Text extraction and results

The automatic approach to text extraction employed in the case of Clearnet sources was based on a proprietary web scraper developed in the Python programming language. The script was being adjusted individually for each website. It employed three modules, namely – *Requests: HTTP for Humans* (for handling HTTP requests), *Beautiful Soup* (HTML/XML parser), and *re* (for enabling the usage of regular expressions). The results of both manual and automatic data extraction amounted to 3337 full texts stored in the *csv* format.

3.2 Annotations

The subsequent step in the process of evaluation set preparation was the annotation of the linguistic data gathered in the previous steps in compliance with the afore-established entity categorization. From the texts collected in the previous steps, a subset of 450 texts was extracted and divided into packages of ten full texts each. The number of 450 texts was primarily due to the limited time commitment of the annotators. Given the availability of annotators and the estimated time needed to tag one entire text file, it was decided to entrust each annotator with tagging 50 full texts in a month, giving an average of one file annotated per week. Each bundle was saved in the *txt* format and labeled with a unique sequence number. The first month was intended to be an introductory period, as there was provision for students involved in annotation to continue their involvement in the project after the first month. However, due to the insufficient annotators willing to do so, the work had to be terminated at 450 texts.

Each of the nine annotators involved was a third-year student of Linguistics and Information Science at Adam Mickiewicz University. The annotators were provided with two extensive text files covering the guidelines for the annotation. Additionally, an introductory meeting was organized to train the annotators and clarify matters of concern in the Q&A format. Due to the aforementioned linguistic background of the students involved, it was possible to commence the operation swiftly after conducting the introductory meeting. Annotators were in constant contact with the coordinator throughout the entire process.

The workspace was organized in the cloud. Every text bundle was saved in the *txt* format and labeled with a unique sequence number. Each annotator was given their own identifier with a structure corresponding to "*A*<No.>" (e.g., *A1, A2,* etc.). A file was created showing the text allocation of each annotator. After completing the annotation of a given text bundle, an annotator was to place the file in a designated folder and mark the work progress in a designated spreadsheet.

[1] https://www.torproject.org/

[2] https://itcontent.eu/aktywne-strony-tor/

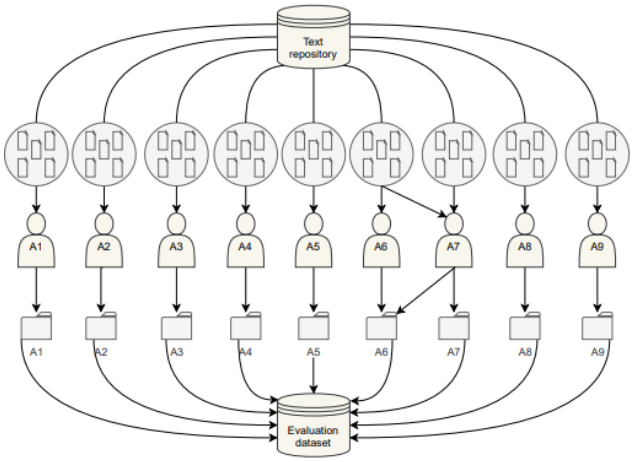

Figure 1: Graphic representation of the annotation process.

The annotators were instructed to mark the beginning and end of each identified named entity and to mark its category. Entities of different categories could be nested.
One subset (*A6*) has been annotated by two independent annotators. Figure 1 presents the scheme of conducted annotation in a concise way.

3.3 Resulting Datasets

450 collected texts were split into fragments according to line breaks. Each fragment consists of one to several sentences or sentences equivalents. As a result, we obtained a set of 6240 annotated text fragments. We will refer to this whole set as *A_all*. Its subset *A6*, annotated by two independent annotators, contained 554 fragments. We used it to evaluate the quality of the manual annotation, as described in Section 5. Additionally, in some experiments, we used sets *A_train* and *A_aug*. The set *A_train* was created as the difference of sets *A_all* and *A6* (i.e., *A_train* := *A_all* – *A6*). The set *A_aug* was created from *A_train* using data augmentation techniques, as described in Section 4.3. The details of the datasets are summarized in Table 1.

Dataset			***A_all***	***A6***	***A_train***	***A_aug***
Number of texts			450	50	400	N/A
Number of text fragments			6240	554	5686	34960
Num. of sentences			6674	637	6037	41955
Number of words			44391	5070	39321	468708
Number of entity instances			11695	1949	9746	75891
	Identifier		398	32	366	1446
	Object		4658	807	3851	34094
		Person	79	4	75	678
		Thing	4579	803	3776	33416
	Action		1188	192	996	9630
	Organization		163	0	163	518
	Location		124	43	81	1408
	Time		320	38	282	3618
	Measure		3020	662	2358	12495
	Description		1824	175	1649	12682

Table 1: The details of the collected datasets.

4. Named Entity Recognition

4.1 System Overview

The named entity recognition system described in this paper is a part of the Context module, developed as a part of the AISearcher software (Demenko et al., 2022). The purpose of the AISearcher system is to support the operation of the Polish Border Guard by facilitating the analysis of Internet resources in terms of crime-related contents, using natural language processing and artificial intelligence.
AISearcher system consists of several modules. In brief, the usage scenario for collecting and analyzing Web texts is the following. First, the user initiates the search, entering a query in a source language (e.g., Polish). The Query Expansion module expands the search term with synonymic expressions. The expanded query is translated to target languages (Russian, Ukrainian, Belorussian) by the Translator module (Nowakowski and Jassem, 2021) and entered into search engines. The search results are translated back to the source language. Then, the Context module is responsible for multi-layer linguistic analysis of the collected texts. The translated and analyzed results are presented to the user. The crucial part of the Context module is the named entity recognition submodule, accountable for the semantic analysis of texts.
The Context module has been developed as a RESTful Web service built upon the Flask web framework[3] to facilitate integration with other AISearcher modules. We also used the Sacred tool (Greff et al., 2017) to manage experiments and the Gonito platform (Graliński et al., 2016) for comparing experiment results.
The Context's NER submodule has a modular structure that allows for testing various NER algorithms, as well as creating ensembles and buckets of multiple algorithm variants. Algorithms are encapsulated in classes following the principles of object-oriented programming so different classes can use the same utility tools, e.g., POS tagger, lemmatizer, or spellchecker.
For POS tagging, we use the Multilingual Universal Part-of-Speech Tagging model (flair/upos-multi-fast) – a multilingual model based on Flair embeddings (Akbik et al., 2018) and LSTM-CRF. For morphological analysis, we use the Morfeusz morphological analyzer for Polish (Kieraś and Woliński, 2017). For spellchecking we use the GNU Aspell spellchecker[4].
We implemented various NER algorithms, both lexicon/rule-based and machine-learning-based, to see which one best performs.

4.2 Approaches Based on Lexicons and Rules

Our rule-based algorithm for recognizing named entities uses carefully prepared lexicons and hand-crafted regular expressions. It also uses hand-crafted rules to assign confidence scores for different entities. These scores are further used in the disambiguation procedure.

4.2.1 Lexicons

For the lexicon-based approach, we prepared domain vocabulary lexicons, as described by Jankowska, Pieniowski, and Demenko (2022). The lexicons contain 3135 lexical units related to different kinds of criminal activity, including the illegal trade of drugs, alcohol,

[3] https://flask.palletsprojects.com

[4] http://aspell.net

cigarettes, cars, machines, weapons, and explosives, as well as document forgery, human trafficking, and sexual offenses. The identified lexical units are categorized according to the classification described in Section 2.1.
The lexicon files are not used by the algorithm as-is but are pre-processed first. We distinguish several kinds of source lexicon files that are treated somewhat differently:

- Most lexicons are just lists of expressions that are to be marked as a particular category. Each of these word lists is assigned a specific category and a specific confidence value.
- Some lexicon files have a frequency value assigned to each contained expression. E.g. the lexicon of towns is a list of town names and their population. A confidence score is calculated based on the population value according to the following rule: the larger population, the more likely the town name should be recognized by the algorithm. A similar method is used for lexicons of given names and family names.
- Some lexicons include inflected terms. The confidence score is calculated taking into account the term's morphological form. For example, nouns in nominative are given a higher score than nouns in other cases.
- There are also lexicons of ambiguities. These files contain expressions that can be interpreted differently depending on the context, e.g., *Warszawa* – 'the city of Warsaw' – should be recognized as *location*, but *Warszawa* – a car brand – should be recognized as *object–thing*.

Multi-word expressions in the source lexicon files are treated specially. A dedicated multi-lexicon object is created for them, taking into account the dependencies between the constituent words of the expressions. This is done because the process of matching the multi-word expressions should take into account the proper inflection.
The street names lexicon file is also treated specially. During its pre-processing, various common variants of address expressions are added to the target lexicon object, e.g., locative case.
The pre-processing of disambiguation lexicon files consists of creating a dedicated disambiguation dictionary that stores the information about the expression, its category, and the sets of positive and negative context words.
Other lexicons are pre-processed by adding the relevant terms and, if necessary, their inflected forms to the so-called pre-lexicon. Then, to speed up the lexicon lookup procedure, this pre-lexicon is converted to an automaton using the pyahocorasick[5] implementation of the Aho-Corasick algorithm (Aho and Corasick, 1975). Finally, all the pre-processed lexicons and the automaton are dumped onto the disk to accelerate the process of loading lexicons when the service starts.
Other lexical resources used in the system are:

- the word frequency list created from the one-million-word subcorpus of the National Corpus of Polish (National Corpus of Polish, 2012; Przepiórkowski et al., 2012) for disambiguation purposes,
- the stop words list[6].

They are also stored on the disk.

4.2.2 Regular Expressions

Hand-crafted regular expressions are used to recognize entities belonging to the following categories:

- identifier: e-mail addresses[7], URLs[8], phone numbers[9],
- location: postal addresses[10],
- time: date and time expressions,
- measure: expressions of numbers with units of measurement and currencies, percentages[11].

4.2.3 Recognizing Named Entities with Rules

In our rule-based named entity recognizer, named entities are recognized in three ways: from regular expressions, the automaton, the lexicons. Recognizing entities from regular expressions is straightforward: a text fragment is marked as a particular named entity category if it matches the corresponding regular expression. Identifying entities from the automaton is similar, but it takes into account word boundaries because the automaton only deals with whole-word phrases. Recognizing entities directly from the lexicons consists of iterating over the lexicon entries (one-word lexicon, multi-word lexicon, and disambiguation lexicon), using the lemmatizer to check for inflected forms, checking grammatical agreement for multi-word entries, checking context for disambiguated entries, and prioritizing obtained matches with confidence scores.
The results are then filtered, rejecting stop words and one-letter words. For some entity types, a recognized entity is rejected if the frequency list indicates that it should be considered a common word rather than a proper name.
Finally, the results from all three sources are gathered and disambiguated before being returned to the user.

4.3 Machine-learning-based Approaches

In addition to the rule-based named entity recognizer, we decided to train a neural network model. To achieve this goal, we needed a large set of text fragments annotated with categories described in Section 2.1. This means that none of the publicly available general-purpose NER datasets would meet our needs.
Due to these constraints, we decided to use the text fragments from the *A_all* set, except those contained in the *A6* subset. We will refer to this dataset as *A_train* (i.e., *A_train* := *A_all* – *A6*). The dataset *A_train* contains 5686 annotated text fragments.
Furthermore, we created an additional dataset using data augmentation techniques (Shorten and Khoshgoftaar, 2019). For this purpose, we used the Query Expansion

[5] https://github.com/WojciechMula/pyahocorasick
[6] from https://github.com/stopwords-iso/stopwords-iso
[7] `([a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+)[.,!?*)]?`
[8] `((https?://)?(www[.])?[a-zA-Z0-9.-]+[.][a-z]{2,5})`
[9] `([+]?[0-9- ]{9,15})[.,!?*)]?`
[10] `([aApPuU]l[.]( [A-ZĄĆĘŁŃÓŚŹŻ][a-ząćęłńóśźż]+)+( [1-9][0-9]*[a-zA-Z]?(([\/]| m[.]? ?)[1-9][0-9]*)?)?)[.,!?*)]?`
[11] `([+-]?[0-9]+([.,][0-9]+)?[ ]?%)`

module of the AISearcher system (Nowakowski and Jassem, 2021). This way, we obtained a set of 34960 annotated text fragments. We will refer to this dataset as *A_aug*.

The neural network architecture is based on the sequence tagger implemented in the Flair framework (Akbik et al., 2019). The input layer is followed by a stack of Flair contextual string embeddings (Akbik, Blythe, and Vollgraf, 2018) for Polish (from both forward and backward language models). The hidden layer size is 512. Additionally, a conditional random field (Lafferty, McCallum, and Pereira, 2001) is used to obtain predictions from the network because the research shows that CRFs are useful in NER-like tasks (Settles, 2004).

This way, we built two neural network models: one trained on the *A_train* set and the other trained on the *A_aug* set.

4.4 Statistical Approach

We experimented with using some text statistics from the *A_train* set to improve the rule-based recognizer. This way, we developed a series of "statistically adjusted" rule-based models. The models are parameterized by two parameters: g_t ("general feasibility threshold") and s_t ("specific feasibility threshold").

We gathered the statistics from the annotated A_train corpus: for every token (wordform) and its lemma in the corpus, we counted how many times it was marked as a named entity of each category. The file with these statistics is used to filter out the output from the rule-based entity recognizer the following way.

For every returned entity (w, e), where w denotes a word and e denotes the entity category, two values are calculated: the "general feasibility" $g(w)$, and the "specific feasibility" $s(w, e)$, defined as:

$$g(w) := \frac{\sum_{e' \in E} c(w, e') - c(w, \emptyset)}{\sum_{e' \in E} c(w, e')},$$

$$s(w, e) := \frac{c(w, e)}{\sum_{e' \in E} c(w, e')},$$

where $c(w, e)$ denotes the number of occurrences of word w annotated as category e in the corpus, $c(w, \emptyset)$ denotes the number of occurrences of word w not marked as a named entity in the corpus, and E denotes the set of all entity categories and $\emptyset$. An entity (w, e) is rejected if $g(w) < g_t$ or $s(w, e) < s_t$.

4.5 Ensemble Models

An ensemble of models is a classifier that combines individual predictions of constituent models in some way (Dietterich, 2000; Dzeroski and Zenko, 2002). Ensembles are created to benefit from the advantages of constituent classifiers.

Our experimental setup allows for building ensembles of models in two ways:

- A bucket classifier uses different models for different named entity categories. We expect that such a bucket model will better reflect the variety of entity categories.
- A weighted ensemble is a classifier that gathers results from constituent models and gives them different weights. These weights are multiplied by obtained confidence scores. The resulting scores are used to prioritize and disambiguate the results by rejecting lower-score entities.

Because ensemble models are implemented so that they all inherit from a model base class, they can be further combined, creating mixed ensembles, e.g., weighted ensembles of buckets.

5. Evaluation

For evaluation, we used the multi-label precision p, recall r and F_β-score metric defined as follows:

$$p = \frac{|T \cap P|}{|P|},$$

$$r = \frac{|T \cap P|}{|T|},$$

$$F_\beta = (1 + \beta^2) \cdot \frac{p \cdot r}{(\beta^2 \cdot p) + r},$$

where P and T are sets of predicted labels and true labels, respectively. The F_1-score is the harmonic mean of the precision and recall. If $\beta < 1$, then the F_β-score values the precision more than the recall. In addition to the widely used F_1-score, we chose the $F_{0.5}$-score for evaluation because we expected users to be more concerned with precision than recall.

We evaluated many models and their combinations (ensembles) and modifications. For comparison, we used the Nerf general-purpose named entity recognizer for Polish (Waszczuk et al., 2013). Nerf is a statistical NER based on linear-chain conditional random fields.

Because the *A6* set has been annotated by two independent annotators, we used this to evaluate the quality of manual annotation, calculating the multi-label F1-score of one annotation versus the other annotation.

Table 2 shows evaluation results for selected classifiers, metrics, and test sets.

Classifier	F_1-score on	
	A6	***A_all***
nerf	0.00512	0.01670
rules	0.45998	0.20874
rules + multiword agreement	0.45904	0.20830
rules + spellcheck	0.39489	0.17865
rules + spellcheck + multiword agreement	0.39419	0.17797
rules + spellcheck + threshold 0.1	**0.48718**	0.21943
rules + spellcheck + threshold 0.3	0.48020	**0.22331**
rules + spellcheck + multiword agreement + threshold 0.2	0.48425	0.22119
bucket 1 (rule-based only)	0.47263	0.22128
bucket 2 (rule-based only)	0.45319	0.21482
rules + statistics (0.5, 0.3)	0.49059	N/A
rules + spellcheck + threshold 0.1 + statistics (0.5, 0.3)	0.51383	N/A
neural	0.37364	N/A
neural augmented	0.35335	N/A
bucket 3	**0.52232**	N/A
ensemble 1	0.52100	N/A
manual	*0.62138*	N/A

Table 2: Multi-label F1-score on both test sets for different classifiers

Classifiers trained on the *A_train* set or the *A_aug* set were not evaluated on the *A_all* set but only on the *A6* set. Selected classifiers are:

- **nerf** – the Nerf tool described above,
- **rules** – the basic rule-based classifier, as described in Section 4.2, but without spelling correction or checking the grammatical agreement of multi-word named entities,
- **rules + multiword agreement** – the rule-based classifier with checking the grammatical agreement of multi-word named entities,
- **rules + spellcheck** – the rule-based classifier with spelling correction,
- **rules + spellcheck + multiword agreement** – the rule-based classifier with a spelling correction and checking the grammatical agreement of multi-word named entities,
- **rules + threshold *t*** – rule-based classifier but results with confidence *c* below given threshold (i.e., $c < t$) are filtered out,
- **rules + statistics (g_t, s_t)** – rule-based classifier enhanced with statistical information, as described in Section 4.4, with general feasibility threshold g_t and specific feasibility threshold s_t,
- **neural** – a neural model trained on the *A_train* set, as described in Section 4.3,
- **neural augmented** – a neural model trained on the A_aug set,
- **bucket 1** – a bucket classifier composed of the following rule-based classifiers:
    - *nerf* for *object–person* and organization categories,
    - *rules + spellcheck + threshold 0.1* for *object–thing* category,
    - *rules + spellcheck + threshold 0.3* for *description* category,
    - *rules + multiword agreement* for *time* and *action* categories,
    - *rules + multiword agreement + threshold 0.4* for *identifier* and *location* categories,
    - *rules + multiword agreement + threshold 0.5* for *measure* category,
- **bucket 2** – a bucket classifier composed of the following rule-based classifiers:
    - *nerf* for *object–person* and organization categories,
    - *rules + multiword agreement + threshold 0.4* for *location*, *object–thing*, *action*, and *time* categories,
    - *rules + multiword agreement + threshold 0.6* for *identifier* and *measure* categories,
    - *rule + spellcheck + multiword agreement + threshold 0.3* for *description* category,
- **bucket 3** – a bucket classifier composed of the following classifiers:
    - *neural* for *identifier* and *object–person* categories,
    - *neural augmented* for *time* category,
    - *rules + statistics (0.5, 0.3)* for *action*, *measure*, and *description* categories,
    - *rules + spellcheck + threshold 0.1 + statistics (0.5, 0.3)* for *object–thing* and *organization* categories,
    - *rules + spellcheck + threshold 0.3* for *location* category,
- **ensemble 1** – a weighted ensemble of *the rules + spellcheck + threshold 0.1 + statistics (0.5, 0.3)* algorithm and the *neural augmented* model, where both constituents are given the same weight,
- **manual** – manual annotation.

Tables 3, 4 and 5 show precision, recall, F1-score and F0.5-score by category for manual annotation and for two select algorithms with relatively high F1-scores: rules + spellcheck + threshold 0.1 and rules + spellcheck + threshold 0.1 + statistics (0.5, 0.3). Categories object – person and object – thing were grouped together as object, and the organization category was not included due to insufficient data in the A6 set. Figures 2, 3 and 4 are precision and recall plots for these three models.

Category	**Prec.**	$F_{0.5}$	F_1	**Recall**
Identifier	0.40678	0.44776	0.52747	0.75000
Object	0.75941	0.66851	0.56674	0.45205
Action	0.53521	0.39916	0.28897	0.19792
Location	0.10811	0.10471	0.10000	0.09302
Time	0.17500	0.17677	0.17949	0.18421
Measure	0.75772	0.67988	0.58910	0.48187
Description	0.20188	0.20935	0.22165	0.24571

Table 3: Multi-label precision, $F_{0.5}$-score, F_1-score, and recall for *rules + spellcheck + threshold 0.1* algorithm.

Category	**Prec.**	$F_{0.5}$	F_1	**Recall**
Identifier	0.52174	0.55556	0.61538	0.75000
Object	0.86375	0.72419	0.58292	0.43990
Action	0.70000	0.48611	0.33333	0.21875
Location	0.22222	0.12658	0.07692	0.04651
Time	0.12500	0.12048	0.11429	0.10526
Measure	0.80151	0.70763	0.60189	0.48187
Description	0.26562	0.24745	0.22442	0.19429

Table 4: Multi-label precision, $F_{0.5}$-score, F_1-score, and recall for *rules + spellcheck + threshold 0.1 + statistics (0.5, 0.3)* algorithm.

Category	**Prec.**	$F_{0.5}$	F_1	**Recall**
Identifier	0.38889	0.33654	0.28000	0.21875
Object	0.73378	0.72073	0.70200	0.67286
Action	0.48000	0.47085	0.45777	0.43750
Location	0.42857	0.21127	0.12000	0.06977
Time	0.46429	0.43333	0.39394	0.34211
Measure	0.91725	0.88787	0.84715	0.78701
Description	0.10302	0.11602	0.14311	0.23429

Table 5: Multi-label precision, $F_{0.5}$-score, F_1-score, and recall for manual annotation. Note that precision scores for the first annotator can be treated as recall scores for the second annotator and vice versa.

6. Discussion

Developing a specialized named entity recognition tool is challenging. The results achieved by Nerf show that general-purpose recognizers are not suitable for highly specialized tasks of this kind. Analytic scores obtained from comparing two independent annotators are much lower than 1.0 - the annotators performed annotation differently despite being given strict guidelines. That is because we are not dealing here with a classical named entity recognition task. Some categories, like identifiers or locations, are named entities, but others, like descriptions or actions, are less precisely defined. The annotators tended to agree on the annotation of measures and objects, but descriptions, identifiers, and, surprisingly, locations turned out to be more ambiguous. The ambiguity of location annotation resulted from the presence of the expressions like *na terenie Warszawy* ('on the territory of Warsaw') or *do 50 km od Sandomierza* ('up to 50 km from Sandomierz'), where it was not clear whether the whole phrase or only the place name should be marked.

These issues were also reflected in the performance of our models. As Tables 2–5 show, our best algorithms (i.e., *bucket 3*, *rules + spellcheck + threshold 0.1* and similar) scored comparably to humans for categories like identifier or description, and only slightly worse for categories like object, action, and measure. The time and location categories proved to be the most challenging.

Rule-based algorithms that filtered out the least confident results achieved one of the highest F_1-scores. The use of statistical data was helpful for this purpose. Neural models performed significantly worse, probably due to a too-small training set. Data augmentation did not mitigate this problem. Carefully designed bucket ensembles proved to be the best.

Another interesting remark is that the analytic measures of NER performance like precision, recall, or F1-score do not always coincide with the users' experience. Although the statistically enhanced models achieved better scores than lexicon-based algorithms, the system users preferred the simple rule-based algorithms. They perceived the statistical and neural models as less "coherent" and "predictable".

We plan to carry out an evaluation on a larger scale in the future. We are systematically updating our named entity recognition algorithms. For this purpose, users – the employees of the Border Guard – constantly collect original materials and provide valuable comments. We analyze these materials and remarks to optimize the entire NER process interactively. We also plan to collect more training data, hopefully allowing us to train more accurate neural models.

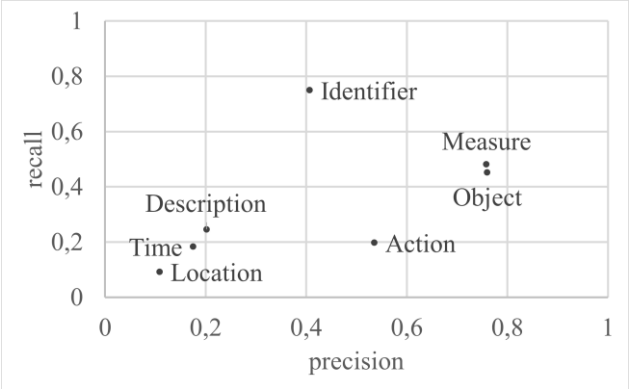

Figure 2: Precision and recall by categories for *rules + spelllcheck + threshold 0.1* algorithm.

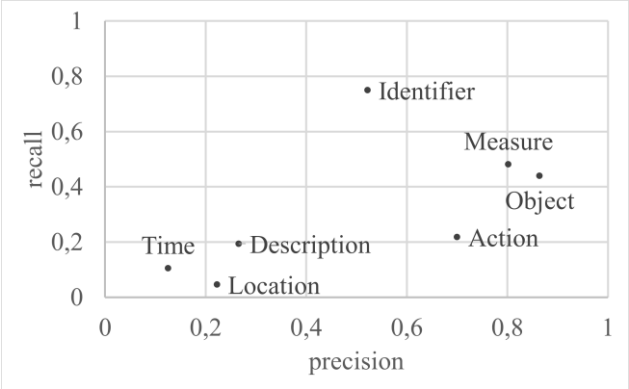

Figure 3: Precision and recall by categories for *rules + spellcheck + threshold 0.1 + statistics (0.5, 0.3)* algorithm.

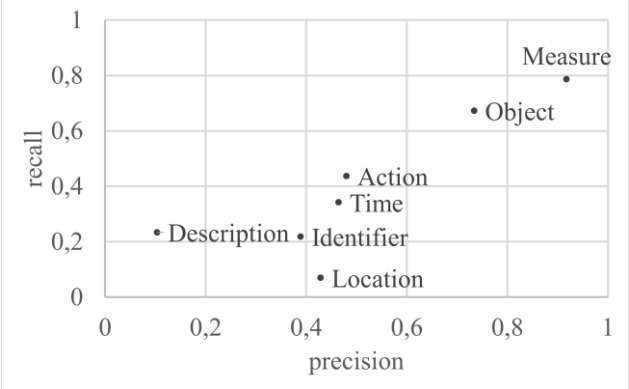

Figure 4: Precision and recall by categories for manual annotation.

7. Acknowledgements

This research was carried out as a part of the project "Advanced analysis of Internet resources to support the detection of criminal groups (AISearcher)", financed by the National Centre for Research and Development (contract number: DOB-BIO9/19/01/2018).

8. Bibliographical References

Aho, A.V., and Corasick, M.J. (1975). Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6), 333-340.

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual String Embeddings for Sequence Labeling. *COLING 2018, 27th International Conference on Computational Linguistics*, 1638–1649.

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 54–59.

Al-Moslmi, T., Gallofré Ocaña, M., Opdahl, A.L., and Veres, C. (2020). Named Entity Extraction for Knowledge Graphs: A Literature Overview. In *IEEE Access*, vol. 8, pp. 32862-32881, DOI: 10.1109/ACCESS.2020.2973928

Baluja, S., Mittal, V.O., and Sukthankar, R. (2000). Applying machine learning for high-performance

named-entity extraction. *Computational Intelligence*, 16(4), 586-595.

Borchmann, Ł., Gretkowski, A.; Graliński, F. (2018). Approaching nested named entity recognition with parallel LSTM-CRFs. In M. Ogrodniczuk and Ł. Kobyliński (Eds.), *Proceedings of the PolEval 2018 Workshop*, Institute of Computer Science, Polish Academy of Science, Warszawa, pp. 63-73.

Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). NYU: Description of the MENE named entity system as used in MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Chau, M., Xu, J.J., and Chen, H. (2002). Extracting meaningful entities from police narrative reports. In *Proceedings of the 2002 Annual National Conference on Digital Government Research* (pp. 1-5).

Demenko, G., Skórzewski, P., Kuczmarski, T., Pieniowski, M. (2022). Linguistic Information Extraction from Text-based Web to Discover Criminal Activity [Unpublished manuscript].

Dietterich, T. (2000). Ensemble methods in machine learning. *In International workshop on multiple classifier systems* (pp. 1-15). Springer, Berlin, Heidelberg.

Dzeroski, S., and Zenko, B. (2002). Is combining classifiers better than selecting the best one? In *ICML* (Vol. 2002, p. 123e30).

Graliński, F., Jaworski, R., Borchmann, Ł., and Wierzchoń, P. (2016). Gonito.net – Open Platform for Research Competition, Cooperation and Reproducibility. In A. Branco, N. Calzolari and K. Choukri (Eds.), *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, pp.13-20.

Greff, K., Klein, A., Chovanec, M., Hutter, F., and Schmidhuber, J. (2017). The Sacred Infrastructure for Computational Research. In *Proceedings of the 15th Python in Science Conference (SciPy 2017)*, Austin, Texas, pp. 49–56.

Jankowska, K., Pieniowski, M., Demenko, G. (2022). Domain Linguistics Resources for discovering criminal activities in Polish Texts [Manuscript submitted for publication].

Kieraś, W., and Woliński, M. (2017). Morfeusz 2 – analizator i generator fleksyjny dla języka polskiego. *Język Polski*, 97(1), 75-83.

Krupka, G., and Hausman, K. (1998). IsoQuest Inc.: description of the NetOwl™ extractor system as used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Marcińczuk, M., Kocoń, J., Oleksy, M. (2017). Liner2 — a Generic Framework for Named Entity Recognition. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, Valencia, Spain, 4 April 2017. Association for Computational Linguistics, pp. 86-91.

Marcińczuk, M., Kocoń, J., Gawor, M. (2018). Recognition of Named Entities for Polish-Comparison of Deep Learning and Conditional Random Fields Approaches. In M. Ogrodniczuk and Ł. Kobyliński (Eds.), *Proceedings of the PolEval 2018 Workshop*, Institute of Computer Science, Polish Academy of Science, Warszawa, pp. 63-73.

Miller, D.R., Leek, T., and Schwartz, R.M. (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 214-221).

Krauz, A. (2017). Mroczna strona internetu – tor niebezpieczna forma cybertechnologii. Dydaktyka informatyki, strony 63-74.

Mider, D. (2019, Listopad 29). Czarny i czerwony rynek w sieci The Onion Router – analiza funkcjonowania darkmarketów. Przegląd Bezpieczeństwa Wewnętrznego, strony 154-190.

Minkov, E., Wang, R.C., and Cohen, W. (2005). Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing* (pp. 443-450).

Nowakowski, A., and Jassem, K. (2021). Detection of criminal texts for the Polish state border guard. arXiv preprint arXiv:2108.10580.

Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B. (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)* (pp. 107-110).

Shorten, C., and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1-48.

Waszczuk, J., Głowińska, K., Savary, A., Przepiórkowski, A., and Lenart, M. (2013). Annotation tools for syntax and named entities in the National Corpus of Polish. *International Journal of Data Mining, Modelling and Management*, 5(2), 103-122.

Witten, I. H., Bray, Z., Mahoui, M., and Teahan, W. J. (1999). Using language models for generic entity extraction. In *Proceedings of the ICML Workshop on Text Mining* (p. 14).

9. Language Resource References

National Corpus of Polish Consortium (2012). National Corpus of Polish, http://nkjp.pl