Feature Extraction for Polish Language Named Entities Recognition in Intelligent Office Assistant

Aleksander Denisiuk
University of Warmia and Mazury
Faculty of Mathematics and Computer Science, Olsztyn, Poland
denisiuk@matman.uwm.edu.pl

Maria Ganzha,
Warsaw University of Technology,Warsaw, Poland,
maria.ganzha@pw.edu.pl

Marcin Paprzycki,
Katrzyna Wasielewska-Michniewska,
Systems Research Institute Polish Academy of Sciences, Warsaw, Poland
firstname.lastname@ibspan.waw.pl

Abstract

The purpose of this contribution is to present a feature extractor that was designed as a part of a Named Entity Recognition (NER) system, which is to be used in a Robotic Process Automation application with a self-learning ability. The NER system has a screen of the user interface as its input, and tries to recognize and categorize all the named entities that can be located within this screen. The set of features that can be extracted from the input, is discussed in the article. The local context features appear to be very important in the considered problem. Experiments show that the entities are recognized with a rate that is satisfactory from the business perspective.

1. Introduction and related works

The aim of Robotic Process Automation (RPA) is creation of software robots that can repeat (simple) tasks performed by workers in application interfaces. Examples of such actions are: logging into systems, moving files, copying and pasting data across different applications [1].

In our research we consider a next generation intelligent office assistant. Besides standard facilities provided by the RPA it is expected to have a self-learning ability. Specifically, it should be able to automatically understand, which user actions belong to which routines, and which routines are good candidates for the automation, as it is outlined, for instance, in [2].

Specifically, the assistant should understand what kind of data is transferred by the user from one application to another. In the developed system we propose to use the Named Entity Recognition (NER) techniques to implement this capability. To the best of our knowledge, this is one of the first attempts to add such capability to the RPA system. At least, none of existing RPA tools analysed in most recent survey [2], has a self-learning capability.

The Named Entity Recognition task consists of locating and categorizing important fragments (called *named entities*) in an unstructured text. Situations when NER is applied include, but are not limited to, business (administration) documents, customer comments, Web pages, and XML files.

A typical NER pipeline is as follows (see, [3, 4] for more details). First, if necessary, given text is turned into a digital format and pre-processed. Pre-processing may involve, for instance, removal of tags, removal of stop-words, stemming, etc. Next, the resulting text is divided into n-grams. Then, each n-gram is converted into a feature vector. Finally, feature vectors are used to train a classifier. Resulting classifier is then used to categorize n-grams appearing in a "production system".

Currently, NER-related methods are actively developed in the context of various applications. They are used, for instance, in Information Retrieval [5, 6], Question Answering [7, 8], or Machine Translation [9, 10]. It is worth noting that many works adapt general NER methods to work with national languages [11, 12], including Polish [13, 14, 15].

Here, let us observe that, in the RPA context, the only available output from *any* application that the (RPA) assistant is going to interact with is a screenshot. So, the NER system, instead of a text document, has an image as its input data. Obviously, there exist hybrid RPA systems where screen-originating data is augmented by old-fashioned text-data, but the focus of our work is limited to the pure RPA scenario, where no such data augmentation/hybridization takes place.

On other hand, note that the data presented within the application screen is often organized in a structured way, e.g. in a table: by rows, or by columns, with appropriate headings. However, entities of interest can also materialize (within the interface) as unformatted text, preceded by a corresponding label. Moreover, the specific format of this structure is likely to differ between applications. Furthermore, it can change after release of the next version of the same application. Hence, to create a general NER system that is stable with respect to changes of the interface, capabilities of the

feature extraction sub-task become extremely important.

The purpose of this contribution is to show, on the level of *proof-of-concept*, the potential of the NER techniques that are to be used in the RPA context.

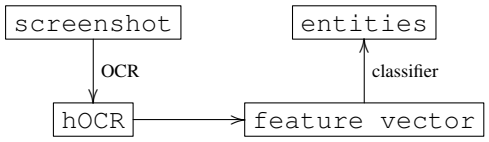

Figure 1. NER pipeline

The Figure 1 shows the adopted NER pipeline. We assume that any OCR software that produces the standard hOCR format [16], can be used as the first step of the pipeline. Moreover, any classifier can be used in the recognition step. So, in this work we concentrate on the feature extraction stage, which is known to be at least as important as the choice of machine learning algorithms in the NER task [3]. Specifically, in Section 3 we introduce a new set of features that describe a local context of an n-gram. These features are screenshot-specific and, often, are not present or lost in the general NER algorithms. The results of our tests show that the local context is very important for recognition and categorization of the entity. See, also, recently published paper [17], where importance of the local context, in the problem of gender recognition of named entities, was considered.

As a model example we consider extraction of information about business organizations registered in Poland. However, the proposed approach can be easily adopted to any other semantic domain, e.g. to medical data, student exam grades, etc. In fact, in the considered algorithms, the only domain-specific aspects are the set of entities and the extracted rule-based feature, `isa`, that reflects the internal format of the entity (see, Section 3 for more details).

The same remark concerns the language. Besides the language-specific set of entities and rules for the `isa` feature, we used a list of Polish stop words [18] and a Polish stemmer [19]. However, the stemmer and the list of stop words can be replaced by comparable tools for any other language. Moreover, the rules for the `isa` feature can be formulated for any language. In this way, the proposed approach is language independent. Nevertheless, we admit that the results of applied machine learning are very likely to differ considerably for each language. Therefore, development of, for instance, Turkish NER, is not a straightforward task of replacing Polish-language modules with Turkish-language modules. Even more interesting and complex would be design of a NER for a business working in a global environment, where documents in multiple languages have to be processed. There, as an additional step, language/country recognition module would have to be added to the pipeline. However, as soon as a country of origin of a given document was established, such document could be directed to a country-specific NER (sub-)pipeline for processing. Hence, the key problem remains the development of a flexible NER module, which is the actual focus of this work.

The remaining parts of the paper are organized as follows. The Section 2 presents the data, specifically we describe attributes of a business organization that are recognized by the considered approach. Next, (in the same Section) the features of n-grams that are extracted from the hOCR data are summarized. The Section 4 summarizes the data preparation (note that, in our work, the experimental setup is rather straightforward). Results of computer experiments are presented and discussed Section 5. Additional discussion and final conclusions are placed, respectively, in Sections 5.5 and 6.

2. Data model

In our work we consider the business organization data as a knowledge domain. As a set of attributes to be recognized we choose ones from the popular public Schema.org dictionary `https://schema.org/Organization` that are commonly present in considered databases. Specifically, we consider the following attributes:

name — the name of the organization.

taxID — the Tax/Fiscal ID of the organization, NIP in Poland. For example, 123-456-32-18.

address — the postal address of the organization. Here, we assume that it consists of the following components:

- **postalCode** — the postal code. For example, 82-300.
- **streetAddress** — the street address. For example, ul. Anhellego 26.
- **addressLocality** — the locality in which the street address is, and which is in the region. For example, Szczecin.
- **addressRegion** — the region in which the locality is, and which is in the country. The voivodeship in Poland. For example, Zachodniopomorskie.

As the input data we used selected Internet sources, instead of a data generated by the actual office software.

This was done for two reasons. First, we wanted to be sure that the available data is correctly labelled (by comparing the result with the original data input). Second, the only way to generate data in-house was to use data originating from processing of actual business documents, which was not open to reporting. Therefore, as the source of the data we used screenshots originating from three different sources:

1. Regon Internet Database — official database of business organizations registered in Poland. It is maintained by the Statistics Poland. In what follows we will refer to that data as *400 series*. The main address of this dataset is `https://wyszukiwarkaregon.stat.gov.pl/appBIR/index.aspx`

2. Rejestr.io — alternative database of Polish business organizations maintained by the Foundation ePaństwo. In what follows we will refer to that data as *500 series*. It is available at `https://rejestr.io/`

3. Arbitrarily chosen web pages, containing basic information about Polish business organizations, mainly the organizational home pages.

Figures 2 and 3 show sample screenshots from the 400 series and the 500 series, correspondingly. Both screenshots were cut-down for presentation purposes. Let us mention some differences that potentially can cause problems in the named entity recognition process. The 400 series splits address into four blocks: street, region, city, and postal code, with separate labels. However, the address in the 500 series data is given in a single continuous fragment: street, postal code, city with one label ("Adres siedziby") placed above the data, while the region is not present at all. Note that the order of entities, represented on the screens/pages, is also different. Moreover, the organization name (and selected other information) in the 500 series is all uppercase, while in the 400 series, it is not.

Figure 5 shows an example of column-major screenshot from our data set. Note that the company name and the street address are written in two lines, contrary to the screenshots from Figures 2 and 3. Obviously, this difference can bring additional difficulty to the named entity recognition process.

The combination of data from the 400 series and the 500 series was used for both training a classifier, and for tests. The third data set was used only for training. This data set was found to be necessary to avoid over-fitting of the classifier. Such over-fitting resulted from training using only one, systematically structured data set.

All the screenshots from all data sets were converted into hOCR format with the Tesseract software, and correspondent entities were manually marked in the JSON format (see, Figure 4).

3. Features extraction

Let us now summarize the process of feature extraction that was applied in our work.

3.1. N-grams

The first step in feature extraction is dividing the document into n-grams. Here, let us note that, despite the fact that the Tesseract is known to have problems with recognition of data from the tables [20], it turned out that its segmentation abilities were good enough for our purposes. In our work, we used the fact that the hOCR standard [16] recognizes blocks of two levels (see, [16] for more details):

- level 1 – `ocr_blockquote`, `ocr_display`, `ocr_par` – an analog of a paragraph, in the traditional typesetting.
- level 2 – `ocr_line`, `ocr_header`, `ocr_caption`, `ocr_textfloat` – an analog of a line in a paragraph, in the traditional typesetting.

This allows one to construct n-grams from singe words (hOCR `ocrx_word` elements) from (correctly recognized) blocks of level 1. That is, in the proposed approach, we do not combine words from different paragraphs into an n-gram. This is based on the nature of our data, in which different paragraphs represent "different content". Additionally, we have restricted the maximal length of the n-gram to 11 (which represents the maximal observed length of organization name in our data). However, this is just a technical issue, again, related to our specific application. This parameter can be adjusted according to the needs of any other NER-based application.

3.2. Features

The features that are used commonly in the NER task are systematised in surveys [3, 4]. We chose the features that are in our opinion appropriate in the NER task in the RPA context, and introduced some new features that catch *a local context* of an n-gram (described in details at the end of Section 3.4). Note that these new features are not present in standard NER setup.

Observe also that some commonly used features are not applicable to the NER considered here. For

INFORMACJE PODSTAWOWE	
REGON	361190802
NIP	7881919929
status NIP	
imię	HUBERT
drugie imię	MICHAŁ
nazwisko	MACIEJEWICZ
data wpisu do REGON	2015-04-07
data skreślenia z rejestru REGON	
kod i nazwa podstawowej formy prawnej	9 - OSOBA FIZYCZNA PROWADZĄCA DZIAŁALNOŚĆ GOSPODARCZĄ
kod i nazwa szczególnej formy prawnej	099 - OSOBY FIZYCZNE PROWADZĄCE DZIAŁALNOŚĆ GOSPODARCZĄ
kod i nazwa formy własności	214 - WŁASNOŚĆ KRAJOWYCH OSÓB FIZYCZNYCH

DZIAŁALNOŚĆ GOSPODARCZA PODLEGAJĄCA WPISOWI DO CEIDG

INFORMACJE PODSTAWOWE	
nazwa	Tłumaczenia Językowe Hubert Maciejewicz
organ rejestrowy	MINISTER ROZWOJU
rodzaj rejestru lub ewidencji	CENTRALNA EWIDENCJA I INFORMACJA O DZIAŁALNOŚCI GOSPODARCZEJ
numer w rejestrze ewidencji	

ADRES SIEDZIBY	
kraj	POLSKA
województwo	WIELKOPOLSKIE
powiat	nowotomyski
gmina	Nowy Tomyśl
miejscowość	Nowy Tomyśl
ulica	ul. Wypoczynkowa
nr nieruchomości	50
nr lokalu	4
kod pocztowy	64-300

Figure 2. A typical screenshot from 400 series

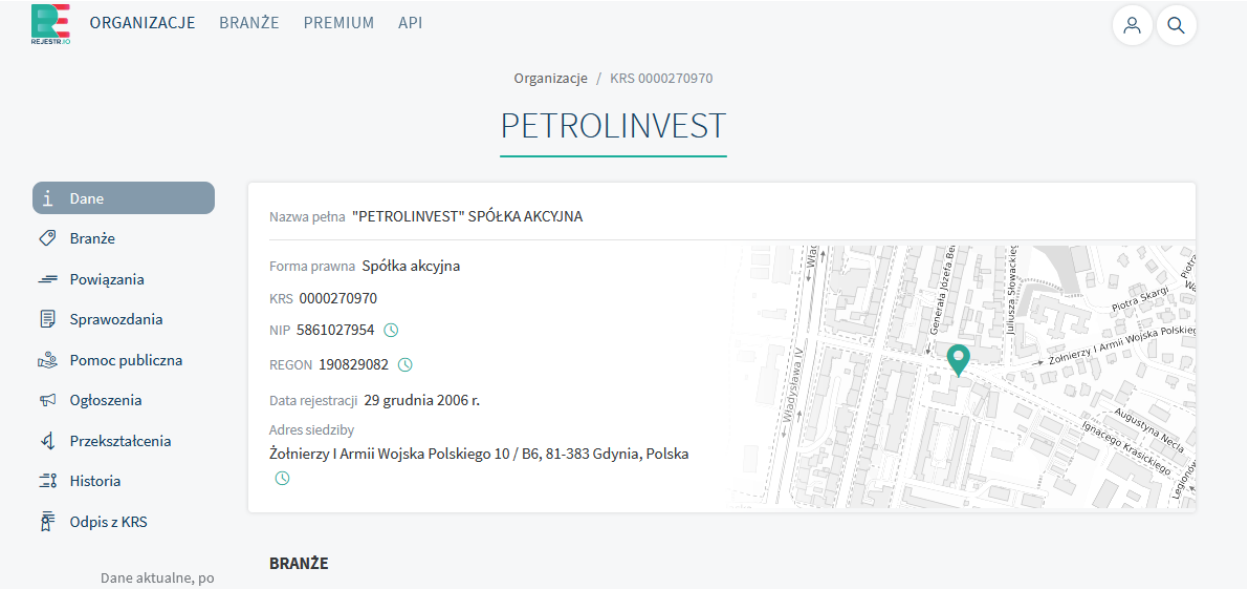

Figure 3. A typical screenshot from 500 series

```
{
 "@context": "https://schema.org",
 "@type": "Organization",
 "address": {
  "@type": "PostalAddress",
   "postalCode": {
     "value": "71-037",
     "elemId": ["word_1_172"]
   },
   "addressRegion": {
     "value": "ZACHODNIOPOMORSKIE",
     "elemId": ["word_1_157"]
   },
   "addressLocality": {
     "value": "Szczecin",
     "elemId": ["word_1_161"]
   },
   "streetAddress": {
     "value": "ul. Anhellego 26",
     "elemId": ["word_1_163",
     "word_1_164", "word_1_165"]
   }
 },
 "name": {
  "value": "ISO-BUD - KAZIMIERZ KARPOWICZ",
  "elemId": ["word_1_125",
  "word_1_126", "word_1_127", "word_1_128"]
 },
 "taxID": {
   "value": "8521848665",
   "elemId": ["word_1_64"]
 }
}
```


Figure 4. Example of entities markup resulting from data preparation

instance, font size can depend on screen resolution and user preferences. Hence, it is not used in the proposed approach. Overall, we do not believe that this feature is of much use in the case of extraction of business name, address and tax-ID, when RPA software delivers the input images (screenshots).

One can divide the feature space, in the developed NER system, into two parts: (1) *internal format features* and (2) *semantic environment features*. The first group contains features that are directly related to the n-gram itself. The second group captures words/terms that are located "in the semantic neighborhood" of an n-gram. Let us now describe each of them in more detail.

3.3. Internal format features

The internal format features include:

- Numeric features:
  1. the length (in words) of n-gram,
  2. the length (in letters) of n-gram.
- Boolean features:
  1. if the text of n-gram capitalized,
  2. if the text of n-gram is uppercase,
  3. if the text of n-gram is formatted as title.
- Nominal features:
  1. `isa` – the rule-based feature that reflects the internal format of n-gram,
  2. the last character of the n-gram (we distinguish the following categories: letter, digit, and separate category for each punctuation sign).

The nominal feature `isa` represents an internal format of a n-gram. We used the following rules for the considered entities:

postalCode – all postal codes in Poland have a format `XX-XXX`, where `X` is a digit.

taxID – there are three forms of tax ID (NIP) in Poland: `XXX-XX-XX-XXX`, `XXX-XXX-XX-XX`, `XXXXXXXXXX`, where `X` is a digit. Here, the last symbol is a control digit.

name – many company names end with one of the following texts (converted to lowercase): `spółka z ograniczoną odpowiedzialnością`, `sp. z o.o.`, `spółka akcyjna`, `sa`, `spółka komandytowa`.

streetAddress – in Poland, many street names begin with one of the following texts (converted to lowercase): `ul.`, `ulica`, `al.`, `aleja`, `1 maja`, `3 maja`.

addressRegion – administrative division of Poland has 16 voivodeships. We used a standard gazetteer to test if an n-gram is the voivodeship's name.

addressLocality – it is possible to create a gazetteer that contains all Polish localities. However, instead, we used a short list of several large polish cities. As it was mentioned in [21] the usage of huge gazetteers does not significantly improve entity recognition.

The `isa` feature is the only domain and language-specific feature in the proposed algorithm. In the case of other language and/or domain of application, the algorithm can be adapted by generating rules similar to the ones above.

3.4. Context features

In this group, we distinguish the following features:

- Numeric feature:
  1. the number of the first word of n-gram in the level one block (see Section 3.1).

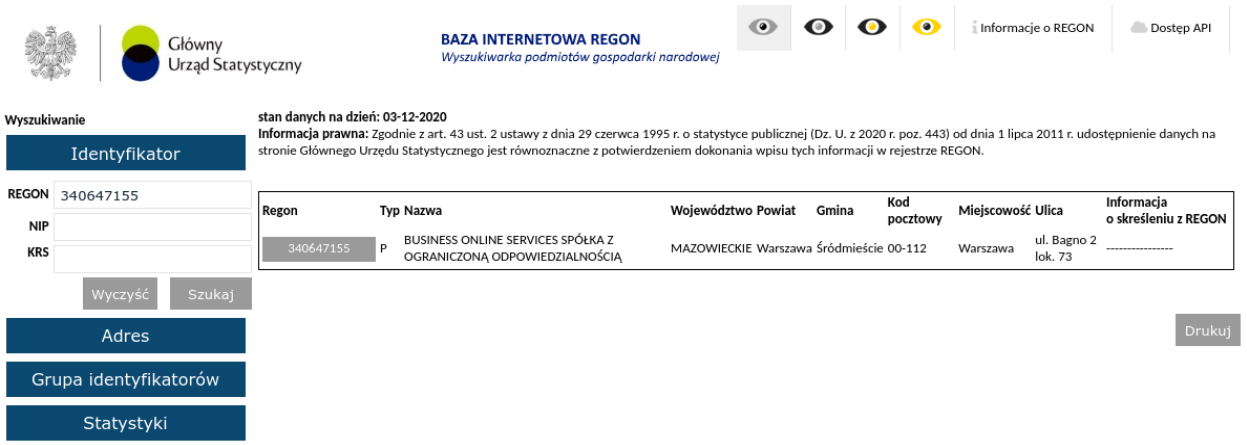

Figure 5. A column-wise screenshot

- Boolean features:
  1. if the n-gram is maximal in the level two of blocks (see Section 3.1),
  2. if n-gram is the last n-gram in level two block.
- Nominal features:
  1. the `isa` property of the word, previous to the n-gram,
  2. the `isa` property of the word, next to the n-gram,
  3. the character previous to the n-gram (we distinguish the following categories: letter, digit, and separate category for each punctuation sign),
  4. all the words previous to the n-gram,
  5. all the words above the n-gram,
  6. all the words to the left of n-gram,
  7. four word that are previous to the n-gram plus four words above plus four word to the left of n-gram.

Combining previous words with words to the left, and words above makes the system independent from the specific form of data representation: column-major, or row-major. Hence, classifier that was trained on row-major data can also recognize entities in column-major data and vice-versa. Here, note that even though labels have mainly length of two-word, we used four words to make the NER process more stable, with respect to possible errors in the OCR software. This was based on results of preliminary experiments.

The last three features describe the *local context* of an n-gram. They turned out to be very important for the NER. In Section 5.4 we report results of experiments where we ran the same test as in the Section 5.1, but instead of these features, we used only lists of word before and after the n-gram to catch the context. According to the literature, these two lists are usually used as a context feature in NER. The reported results show that the performance of standard (simplified) approach is considerably worse.

4. Data preparation

To perform experiments we manually marked 106 screenshots, including 21 from 400 series, and 29 from 500 series, which generated the total of 85 598 n-grams for training and testing. This data is available at: `http://wmii.uwm.edu.pl/~denisjuk/uwm/OAD`.

We used the Tesseract [22] verson 4.1.1 with Polish language recognition feature, as the OCR program in the pipeline (see, Figure 1). We used the following command line parameters:

```
tesseract ... -l pol hocr
```

The output of the Tesseract was the standard hORC file [16], with no further modifications.

Since our main goal was the development of the feature extractor, only the decision tree classifier was used as the last step of the pipeline. This classifier is widely used (and suggested to be used) in NER systems (see [23]). We utilized the standard implementation of decision tree from the Scikit-learn bundle [24], version 0.23.2. We used the default parameters of the constructor: `DecisionTreeClassifier()`. Exploring other possible classifiers will be the content of further work.

The bounding box attribute (`ocrx_word`) of the hOCR elements was used to check if the word is to the left to or above given block. The last four features

were processed using the bag-of-words model. All the words were stemmed with the Stempel Stemmer [19] before being included to the lists. We also used the hashing trick [25, 26] for representation of word lists. The scikit-learn implementation [24] (version 0.23.2) of the hashing trick was used in the experiments. The following parameters were passed to the constructor:

```
FeatureHasher(n_features = 2**20,
        input_type='string' )
```

Let us recall that the input data from the RPA system consists of screenshots of application programs. So, generally, input does not carry any global linguistic semantic. Instead, the impact of local context seems to be more important. So, we considered simple bag of words model at this stage. Investigation of more sophisticated models, i.e. Word2Vec, is planned for the future.

5. Testing

We performed four groups of tests. As a measure of correctness of recognition we used the standard F1 score. This measure is commonly used in evaluation of accuracy of NER approaches (as discussed in [3]). To calculate the average rate, we used the total true positives, false negatives and false positives.

5.1. 20-fold cross-validation test

The first experiment was a 20-fold cross-validation test. We split the data from screenshots randomly into two groups: one for training (80%), another one for testing (20%). The second group contained only data from 400 and 500 series. This reflects the fact that the application of the trained classifier is to be in intelligent assistant that processess data from a computer program interface.

The Figure 6 contains average results of 20 tests. One can see that `addressRegion`, `postalCode` and `taxID` entities are stably recognized with 100% rate. This is mainly due to the rigid structure of the data (re)presentation, and the fact that these entities were labeled with the same text in all the screenshots (see Section 2). The remaining entities also have high recognition rate. It can be stipulated that the recognition rate can be improved by using a more sophisticated classifier (for instance, XGBoost, or random forest). However, in our application, the recognition should be performed in real-time, so the usage of more complicated classifiers may not be feasible. Checking this pathway will become one of future research directions.

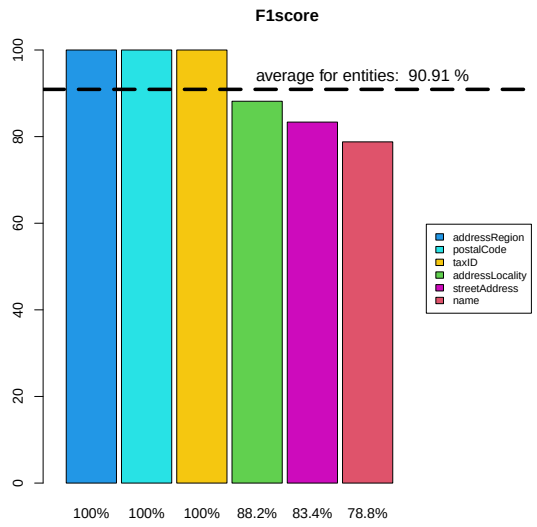

Figure 6. Results of the 20-fold cross-validation test

5.2. Change of the interface

The second set of tests emulates the change of the interface that the intelligent assistant receives the data from. In one experiment we use data from the 400 series for testing while the remaining data was used for training. The second experiment does the same with the 500 series data for testing and the remaining data used for training. In this way, in both experiments, the classifier is tested on screenshots that it has never seen. In both experiments the tests where repeated 20 times (and we report averaged results).

The results of these experiments are depicted in Figures 7 and 8. We see that `addressRegion`, `postalCode` and `taxID` entities are still stably recognized, with 100% success rate (note that `addressRegion` is not present in the 500 series data).

Let us also notice very poor performance of the classification of the `streetAddress` entity in the 500 series data. This was expected. As mentioned in Section 2, in most training data this entity has different format and different label(s).

5.3. Adaptive tests

In this set of experiments we tested ability of the system to adapt itself to change of interface of the user application. Here, let us assume that the user corrects mistakes of the intelligent assistant. Next, after such correction, the NER system is retrained taking into account new knowledge. The main question is: how many corrections should the user make before the

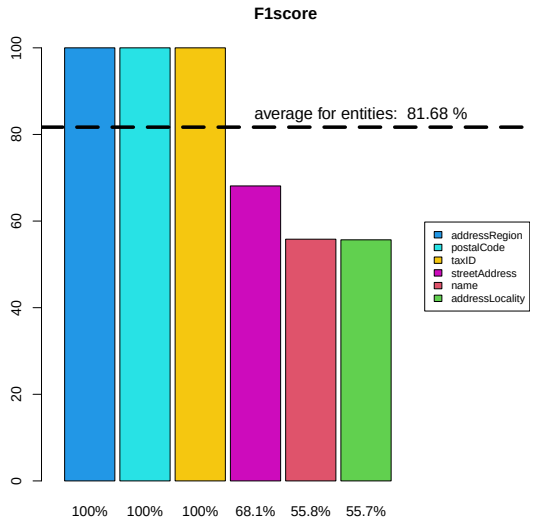

Figure 7. Recognition of entities on 400 series data

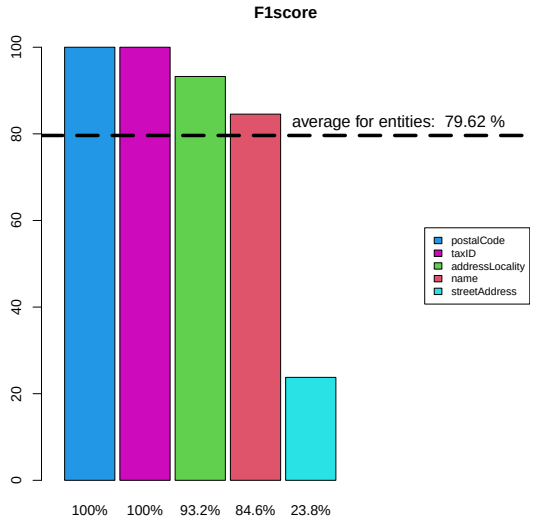

Figure 8. Recognition of entities on 500 series data

system achieves (almost) the previous recognition rate.

In this set of tests, we moved consequently from one to ten randomly chosen screenshots of the series from the test set to the train set and repeated the recognition.

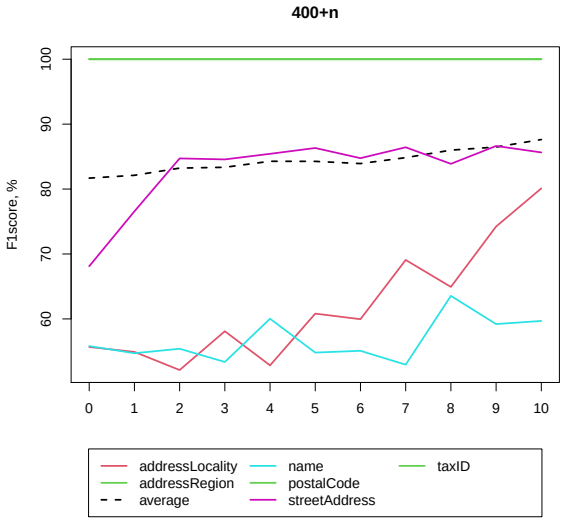

Figure 9. Adaptive test for 400 series data

Figures 9 and 10 represent adaptive tests for data from 400 and 500 series respectively. As in previous experiments, each test was repeated 20 times, and the average results are presented. The system shows a clear adaptability trend. Especially, it is seen for the `streetAddress` feature, which started with a very low recognition rate.

Let us mention that the tests results reflect the difference between address representation in 400 and 500 series screenshots (Figures 2 and 3 respectively). Namely, 500 series screenshots contain data in the "joined form"

```
streetAddress
    + postalCode + addressLocality
```

while 400 series screenshots contain each part of address separately. The first format is commonly used in Poland and is present on most screenshots that do not belong to 400 or 500 series. That is why one can observe significant difference in recognition rate of `addressLocality` and `streetAddress`.

5.4. Recognition without local context feature

Finally, to emphasize the importance of the local context feature (see, Section 3.4), we repeat the 20-fold cross-validation test (Section 5.1) without the local

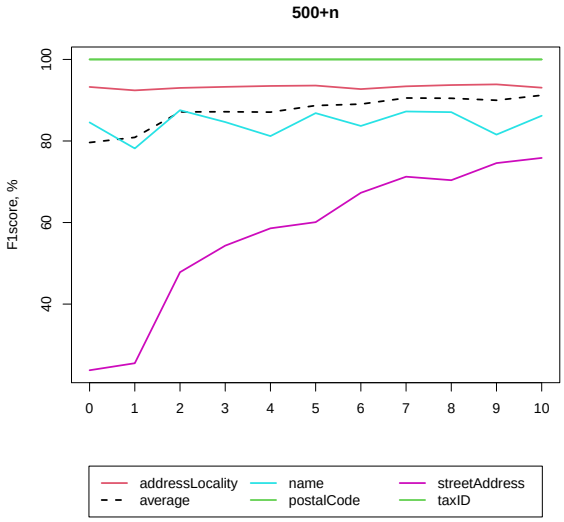

Figure 10. Adaptive test for 500 series data

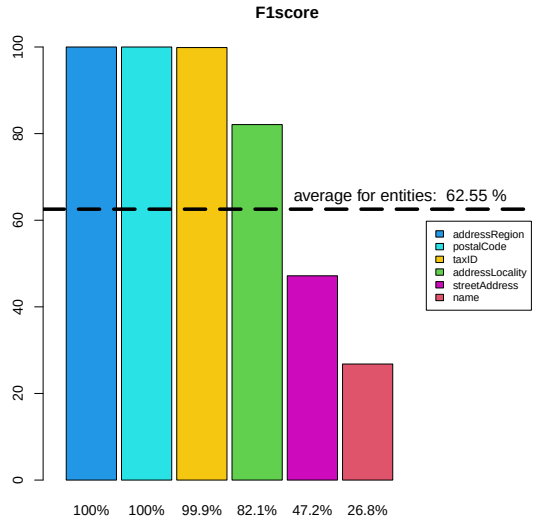

Figure 11. Recognition of entities without local context features

context feature. Instead, as the features, we use the previous and the following words. As noted above, these features are typically used in standard NER to provide context for unstructured cases.

As one can see from Figure 11, the F1 score of the algorithm is considerably lower. The average value for all entities is smaller by about 30%. Even the `taxID`, that has a very rigid structure, and was always correctly recognized, has recognition rate less than 100%.

5.5. Discussion

The presented solution shows a possible application of NER techniques to solve real-life problem of feature extraction. Our goal was to practically recognize attributes that are necessary (and minimal) to describe a business organization. Some of the attributes have rigid structure and naturally their recognition is easier. Those with varying format obtain lower scores, especially after changes of interface. It is possible that the recognition rate can be improved by using a more sophisticated classifier, but such change needs to be balanced with the need for performance in near-real-time. Last set of described experiments proved that inclusion of local context feature significantly improves the F1 score. Note that, in the proposed solution, we tried to be as language and domain agnostic as possible. Dropping this design assumption, might bring about one more possibility to improve performance, but will result in lower adaptability of the solution.

6. Concluding remarks

In the paper we presented a NER system designed for a Robotic Process Automation application with self-learning ability. This system is to deal with Polish language content. It has a screen of the user interface as its input and tries to recognize and categorize all (a'priori predefined) named entities contained within it. We proposed the set of features that can be extracted from the input data. Specifically, we defined the local context featured that are screenshot specific and, often, are not considered in standard general NER systems. Experiments showed that application of local context knowledge is very important in the considered problem.

The future work is planned in the following directions. First, we will test our algorithm in a different knowledge domain(s), e.g. in a medical documentation application. The second task is exploration of other possible classifiers, including the ones using semi-supervised, or unsupervised, learning. Specifically, we plan to adapt a new method of clustering combinations of nominal-continuous data based on automatic metric detection [27]. This method has shorter time of classification in comparison with decision trees, which is important in the RPA context. Next, as mentioned in Section 4, usage of more sophisticated model than a simple bag of words, may improve the results. Finally, we plan to test the impact of resolution and color settings, as well as segmentation parameters on the quality of the OCR step.

7. Acknowledgement

Work of A. Denisiuk, M. Ganzha, and K. Wasielewska-Michniewska has been realized in part within the scope of the "Conducting R&D works for the intelligent OfficeAgent solution" project, which is co-financed by the European Regional Development Fund under the Regional Operational Program of the Mazowieckie Voivodeship 2014-2020.

References

[1] W. M. P. van der Aalst, M. Bichler, and A. Heinzl, "Robotic process automation," *Business & Information Systems Engineering*, vol. 60, no. 4, pp. 269–272, 2018.

[2] S. Agostinelli, A. Marrella, and M. Mecella, "Research challenges for intelligent robotic process automation," in *International Conference on Business Process Management*, pp. 12–18, Springer, 2019.

[3] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, pp. 3–26, Jan. 2007.

[4] B. Mohit, "Named entity recognition," in *Natural Language Processing of Semitic Languages* (I. Zitouni, ed.), pp. 221–245, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[5] M. A. Khalid, V. Jijkoun, and M. de Rijke, "The impact of named entity normalization on information retrieval for question answering," in *Advances in Information Retrieval* (C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. W. White, eds.), (Berlin, Heidelberg), pp. 705–710, Springer Berlin Heidelberg, 2008.

[6] R. More, J. Patil, A. Palaskar, and A. Pawde, "Removing named entities to find precedent legal cases," in *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019* (P. Mehta, P. Rosso, P. Majumder, and M. Mitra, eds.), vol. 2517 of *CEUR Workshop Proceedings*, pp. 13–18, CEUR-WS.org, 2019.

[7] E. Noguera, A. Toral, F. Llopis, and R. Muńoz, "Reducing question answering input data using named entity recognition," in *Proceedings of the 8th International Conference on Text, Speech and Dialogue*, TSD'05, (Berlin, Heidelberg), p. 428–434, Springer-Verlag, 2005.

[8] H. Kilicoglu, A. B. Abacha, Y. Mrabet, K. Roberts, L. Rodriguez, S. Shooshan, and D. Demner-Fushman, "Annotating named entities in consumer health questions," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* (N. C. C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, eds.), (Paris, France), European Language Resources Association (ELRA), may 2016.

[9] B. Babych and A. Hartley, "Improving machine translation quality with automatic named entity recognition," in *Proceedings of the 7th International EAMT Workshop on MT and Other Language Technology Tools, Improving MT through Other Language Technology Tools: Resources and Tools for Building MT*, EAMT '03, (USA), p. 1–8, Association for Computational Linguistics, 2003.

[10] R. Sellami, F. Deffaf, F. Sadat, and L. H. Belguith, "Improved statistical machine translation by cross-linguistic projection of named entities recognition and translation," *Computación y Sistemas*, vol. 19, no. 4, 2015.

[11] R. Al-Rfou, V. Kulkarni, B. Perozzi, and S. Skiena, "POLYGLOT-NER: Massive multilingual named entity recognition," *arXiv preprint arXiv:1410.3791v1*, 2014.

[12] S. Chen, Y. Pei, Z. K. 2, and W. Silamu, "Low-resource named entity recognition via the pre-training model," *Symmetry*, vol. 13, no. 786, pp. 3–26, 2021.

[13] A. Pohl, "Knowledge-based named entity recognition in polish," in *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems* (M. Ganzha, L. Maciaszek, and M. Paprzycki, eds.), pp. 145–151, IEEE, 2013.

[14] K. Wróbel and A. Smywinski-Pohl, "Kner: Named entity recognition for polish," in *Proceedings of the PolEval 2018 Workshop*, pp. 101–108, Institute of Computer Science, Polish Academy of Sciences, 2018.

[15] M. Marcińczuk and A. Wawer, "Named entity recognition for polish," *Poznan Studies in Contemporary Linguistics*, vol. 55, no. 2, pp. 239–269, 2019.

[16] K. Baierer, ed., *hOCR — OCR Workflow and Output embedded in HTML. Living Standard, version 2.1*. http://kba.cloud, 2020.

[17] S. Das and J. H. Paik, "Context-sensitive gender inference of named entities in text," *Inf. Process. Manag.*, vol. 58, no. 1, p. 102423, 2021.

[18] Ranks NL company, "Polish stopwords," 2020.

[19] M. Gawinecki, "Stempel Stemmer, version 1.2.0," 2020.

[20] Tesseract-ocr, *Tesseract User Manual. Improving the quality of the output*, 2021.

[21] A. Mikheev, M. Moens, and C. Grover, "Named entity recognition without gazetteers," in *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, 1999.

[22] Google, "Tesseract, version 4.4.1," 2019.

[23] S. Sekine, "Nyu: Description of the japanese ne system used for met-2," in *Proc. of the Seventh Message Understanding Conference (MUC-7*, 1998.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] J. Moody, "Fast learning in multi-resolution hierarchies," in *Proceedings of the 1st International Conference on Neural Information Processing Systems*, NIPS'88, (Cambridge, MA, USA), p. 29–39, MIT Press, 1988.

[26] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, (New York, NY, USA), p. 1113–1120, Association for Computing Machinery, 2009.

[27] A. Denisiuk and M. Grabowski, "Embedding of the Hamming space into a sphere with weighted quadrance metric and c-means clustering of nominal-continuous data," *Intelligent Data Analysis*, vol. 22, no. 6, pp. 1297–1314, 2018.