THE PROBLEM OF NAMED ENTITIES UNIFICATION BASED ON GEOGRAPHICAL ONTOLOGIES

Talshyn Sarsembayeva
Department of Artificial Intelligence & Big Data
al-Farabi Kazakh National University
Almaty, Kazakhstan
sarsembayeva.talshyn@gmail.com

Madina Mansurova
Department of Artificial Intelligence & Big Data
al-Farabi Kazakh National University
Almaty, Kazakhstan
mansurova.madina@gmail.com

Darya Chikibayeva
Department of Computer science
al-Farabi Kazakh National University
Almaty, Kazakhstan
dashachikibaeva@gmail.com

Dariya Karymsakova
Department of Artificial Intelligence & Big Data
al-Farabi Kazakh National University
Almaty, Kazakhstan
karimsakovadarikosh@gmail.com

***Abstract*— The subject of this research is to develop a system for extracting knowledge from both semi-structured and unstructured data and filling with this system a knowledge base that would provide support for decision-making on any problematic issues. The article deals with the problem of unification of named entities based on geographical ontologies.**

***Keywords*— semi-structured and unstructured data, decision support system, ontology, thesaurus, information extraction, knowledge extraction, knowledge base, machine learning, named entities.**

I. INTRODUCTION

The main purpose of this work is to improve the quality of decision-making by the user based on the analysis of information consolidated and cleared of information noise, extracted automatically from heterogeneous open sources. The extraction of a special type of information from text documents, exactly named entities, along with the extraction of concepts and relationships in the domain, is one of the most important tasks in the construction of intelligent systems that provide conceptual modeling of the domain. This task is especially relevant when developing intelligent decision support systems.

Since the task of extracting named entities is associated with processing large amounts of data, it is efficient to use machine learning methods to solve this problem. However, the variety of classes of extracted entities and their dependence on the subject area is not the only problem that arises by solving this problem. The task of named entity extraction, which is identifying the names found in the text, will not be fully solved without the stage of standardization. Linking each identified name in the text with the concept of ontology / vocabulary is a fundamentally important point for understanding the identified entities.

For the development of an intelligent decision support system in the geoinformation industry, we have proposed a high-level extraction scheme for knowledge extraction from heterogeneous data sources to improve the quality of decision-making, shown in Fig. 1.

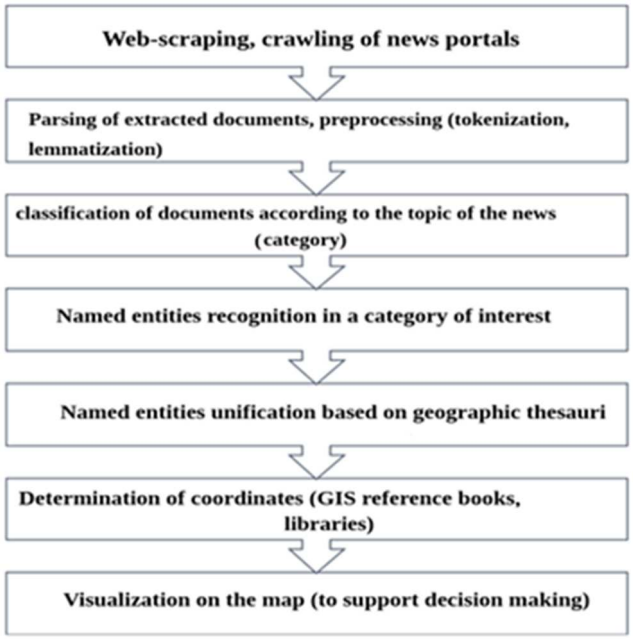

Fig. 1. High-level extraction scheme for knowledge extraction from heterogeneous data sources to improve the quality of decision-making

II. ARCHITECTURE OF THE SYSTEM FOR EXTRACTION OF NAMED ENTITIES BASED ON ONTOLOGY

Analysis of news about emergencies situations (ES) is a relevant part of the economy and society. New information about the ES can be obtained from various sources, including the mass media. Thereby the issue of developing systems capable of automatically collecting, storing, analyzing and structuring news information is relevant. The constant development of new technologies and the emergence of new requirements complicate the development of existing automated systems after obsolescence, which is one of the main problems of digitalization of systems [1]. Thus, an approach that pursues migration from legacy technology and has evolvability properties that allow for system upgrades to meet new requirements and easy migration to newer technologies is needed. Systems that perform named entity extraction using ontologies and linguistic resources, as a rule, operate in a two-step mode:

978-1-6654-2538-4 / 21/31 $. 00 © 2021 IEEE

a) in the first step, such systems scan (read) named entities without specifying an additional information. This is the so-called entity identification step;

b) in a second step, such systems classify, using rule-based methods, the type of named entities previously discovered. This is the so-called step of classifying named entities.

Thus, in the first step the system must analyze each suggestion of the input corpus of the text with the help of a parser or analyzer. The parser should be able to find a sentence that corresponds to the named entities by connecting them to known dictionaries or other linguistic resources. The analyzer must provide information about the recognized concept, information about the term itself, and the fragment of the source text for each identified object. The parser considers the sentence in the order of each token and maybe not correctly recognize the structural entities or consider each part separately. For this reason, the parser is supported by motion "window" which allows to highlight bigrams, trigrams and other multi-component texts.

As a result of this process, the system forms a list of named entities that have not yet been identified as any type.

At the second stage, the system classifies the list of named entities extracted at the parsing stage. For this purpose, a rule-based system has been developed that uses ontologies and annotated knowledge bases, such as DBpedia and others.

The data models describing geographical objects were taken as the object of the research. Ontologies for geographic objects developed within the GeoNames project are described in [2]. The geographic database of these GeoNames contains more than 25 million geographical names and consists of more than 11 million unique features, of which 4.8 million are populated points and 13 million are alternative names. All functions are categorized into one of nine object classes and further subcategorized into one of 645 object codes.

There aren't any ready-made ontologies for geographic objects of the Republic of Kazakhstan. Therefore, the task of designing, constructing and filling in the ontology for geographical objects of Kazakhstan was set

III. EXTRACTION MODEL FOR NAMED ENTITIES

A. An ontology-based extraction model for named entities

The model for extracting geographic names from arbitrary text and the relationship between them based on ontology is built as follows.

− For the input text, perform all the stages of preprocessing: tokenization, expansion of abbreviations, cleaning from stop words, morphological analysis, lemmatization. In this case, the abbreviated words are replaced with unabbreviated meanings using a dictionary of abbreviations templates.

− Label Geographic Locations: Assign word combinations that represent the name of the geographic location, a specific label, or a unique identifier for the geographic location. In this case, the problem of resolving ambiguity may arise. To solve this problem, the method of analyzing the hierarchical relationships of neighboring identified objects in the text is used. Ontology-based named entity extraction systems are very sensitive to the quality of using linguistic resources, dictionaries and knowledge bases . For subject areas that do not have such linguistic resources, these approaches are unsuitable and, accordingly, machine learning-based approaches are required.

B. A random walk-based extraction model for named entities

The purpose of this stage is to create a technology for extracting named entities from news messages based on the random walk method. The following tasks were set to achieve this goal:

1. Carrying out morphological analysis of texts.
2. Modeling texts in the form of a hypergraph and applying the random walk method to extract semantically related words.
3. Creation of a neural network trained to match specific keywords to descriptors.

The first task is associated with preprocessing and morphological analysis of texts for further more objective comparison of models and for unification of input data. The process of "clearing" text from inter-word punctuation, often called trimming, is built into many programming languages, including Python. Splitting into sentences is a less trivial task, due to the specifics of parsing, as well as the fact that the resulting texts often contain punctuation that differs from the original. Nevertheless, a solution was found to work with such cases, the correctness of which was confirmed during experiments. The solution consists in a preliminary morphological analysis, the data of which make it possible to break complex sentences into their simple components, and also divides the text into homogeneous fragments. To bring words to their initial form, a method was chosen for determining parts of speech based on the data of SinTagRus [3] and the support vector machine. The method is based on the assumption that parts of speech can be determined by the endings of words. It is worth noting that here the endings of words are n characters from the end of the word, and not the ending in the morphological sense. Therefore, it was decided to use the pymorphy2 library for morphological analysis, which has already proven itself in text analysis tasks, and the NLTK module [4] for the separation of service parts of speech. The parse function of the pymorphy2 library, receiving a word as input, outputs a list of the most probable characteristics, such as gender, plural or singular, case, part of speech, initial form, sorted by probability. Therefore, choosing the first element of the list, one can obtain the desired characteristics with a given probability and choose an initial form from them. The NLTK module is convenient for separating service parts of speech, it can display them in the form of a list, which allows you to find them in the text by searching for a substring, bypassing the time-consuming morphological analysis. Feature vectorization is currently well automated, so only some clarifications are given here. So, the service parts of speech remain in the corpus, but are specified in the feature vector by much smaller values than the rest. Also, when a certain threshold of the cosine distance is exceeded, the words are combined into a cluster, which corresponds to one component of the vector. This allows taking into account synonyms, as well as optimizing calculations by decreasing the dimension of the feature vector. The cosine distance is calculated based on the similarity of word contexts in texts, where their order is grammatical. This allows you to find words that are close in meaning without resorting to complex analysis. For greater

reliability, when searching for synonyms, the data of morphological analysis are used. Although for the most part, this is a contextless approach, it is possible to train the word2vec model by mapping words to an n-dimensional vector. It can be used to create features.

The use of the random walk method to extract facts was previously described in [5]. The next task is to create a multilayer perceptron model trained to extract linguistic constructions, which include the possible values of the attributes of the named entities of the processed texts. A collection of linguistic constructs and sets of semantically related words will allow training a neural network.

The created neural network makes it possible to extract information according to one pre-selected descriptor, for example, location, giving the names of geographical objects as the final result. In general, a neural network can retrieve information for several descriptors at the same time.

To train the neural network, a training set was built, consisting of feature vectors. For one descriptor, the feature vector was compiled as follows: a window of five words was taken up to the entry point of the element of interest to us in the text of the article and in two words after. Moreover, a dictionary is formed for each descriptor, which is responsible for the presence of the specified word in the dictionary. All features of each descriptor are collected into one "bag of words" and a feature vector is constructed.

The network is trained by presenting each input dataset and then propagating the error. The neural network training algorithm is based on the backpropagation method.

Despite the slight difference between the results of gradient descent and a neural network with hidden layers, the standard deviation is different, which indicates the advantage of a neural network(TABLE 1).

TABLE 1. RESULTS OF EXPERIMENTS WITH DIFFERENT MODELS

Model	Results
Logistic regression	Weighted F1-measure: 0.9 (+/- 0.15) [LogisticRegression(C=100000.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='warn', n_jobs=None, penalty='l2', random_state=None, solver='warn', tol=0.0001, verbose=0, warm_start=False)] Min:0.86 Max0.94
Gradient descent	Weighted F1-measure: 0.93 (+/- 0.2) [SGDClassifier(alpha=0.0001, average=False, class_weight=None, early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True, l1_ratio=0.15, learning_rate='optimal', loss='hinge', max_iter=1000, n_iter_no_change=5, n_jobs=None, penalty='l2', power_t=0.5, random_state=None, shuffle=True, tol=0.001, validation_fraction=0.1, verbose=0, warm_start=False)] Min:0.91 Max0.95
Neural network with hidden layers	Weighted F1-measure: 0.95 (+/- 0.01) [MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, hidden_layer_sizes=(100,), learning_rate='constant', learning_rate_init=0.001, max_iter=200, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=None, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False)] Min:0.94 Max0.95

C. Building an information extraction model based on machine learning methods

The named entity extraction task was based on three named entity extraction algorithms: BERT, bi-LSTM, CRF-baseline.

Bi-LSTM algorithm. To solve the problem of extracting named entities, a model based on the bi-LSTM block with vectorization of characters and words has been developed . The main idea of bi-LSTM is to take into account the sequence of characters not only before the current entry, but also after [6]. Thus, the complete environment is considered. Bi-LSTM consists of a forward-directed LSTM, which considers the sequence of inputs before the current one, and a reverse-directed LSTM, which considers the sequence of inputs after the current one. Then the resulting sequences are concatenated. In this work, the vectorization method was used - one-hot embedding [7]. The purpose of word embeddings is to extract information from a text corpus and associate each of its elements (word / symbol) with a unique numeric vector. Vectorization is one of the approaches to language modeling and representation training in natural language processing, aimed at matching words from a certain dictionary of vectors of a significantly smaller number of words in the dictionary. The theoretical basis for vector representations is a distributive approach to natural language processing, which is a group of methods designed to study the semantic proximity between linguistic units (words, concepts, documents) based on an assessment of the distribution of words in texts. The main tools of distributive analysis are context vectors and co-occurrence matrices [8]. A contextual vector of a word is understood as a vector indicating words with which the given word occurs in the same context [9]. A contextual vector of a document is a vector indicating words that appear in this document. Then the semantic distance between two words or documents is defined as the Euclidean distance or cosine proximity between the corresponding context vectors (Fig. 2)..

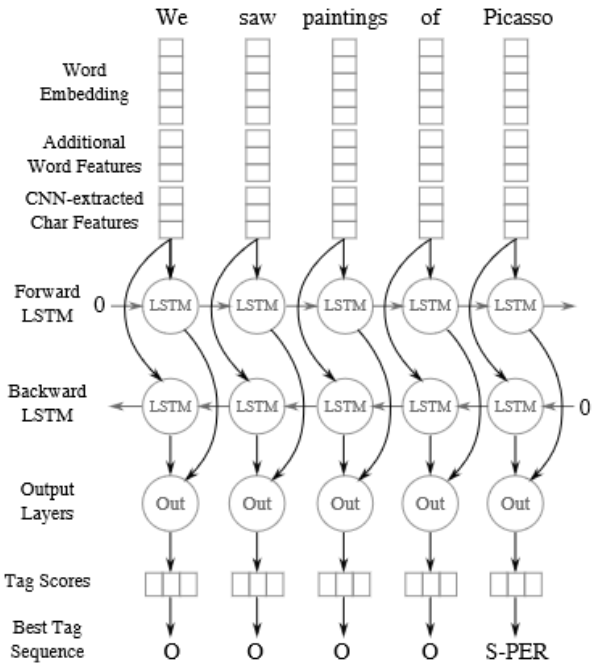

Fig. 2. The model architecture proposed by Jason P.C. Chiu and E. Nichols [10]

The data was marked up using seven tags: PER – PERSON (person, names of people), LOC – LOCATION (location), ORG – ORGANIZATION (organization), B – BEGINNING (beginning, first token of a named entity), I – INSIDE (subsequent (internal) named entity tokens), O – OTHER, a token that is not a named entity.

In recent years, modern models for the named entity extraction problem have been based on pretrained language models. These include BERT, a language representation model developed and pre-trained by Google.

In our system, BERT is used for the task of tagging individual sentences. The model architecture consists of a BERT model followed by a classifier. In this work, BERT was used twice. The first time we use it to represent the offer as tokens, and then BERT is used to get encoded representations of the gaps

Neural network models require large amounts of data, so we decided to benchmark their performance against a method that gives satisfactory results even for small amounts of data. For this purpose, we have chosen conditional random fields, a framework for modeling probabilistic sequences [11], which has been used to create extraction models for named entities in the past [8].

IV. Designing and Building a Geographic Ontology

The stage of design and construction of an ontology indicating classes, data relations, object relations was carried out in the OWL environment. Below are screenshots of the constructed ontology (Fig. 3-5).

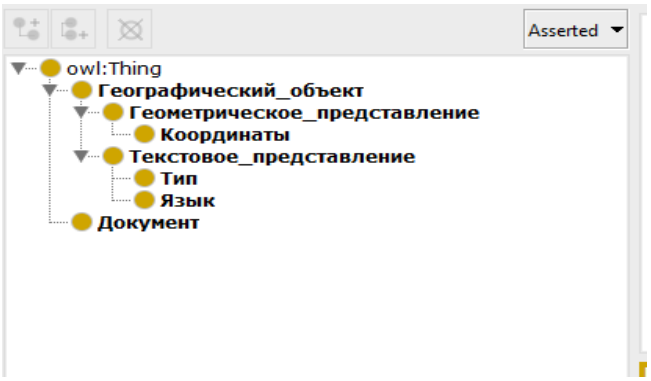

Fig. 3. Hierarchy of ontology classes

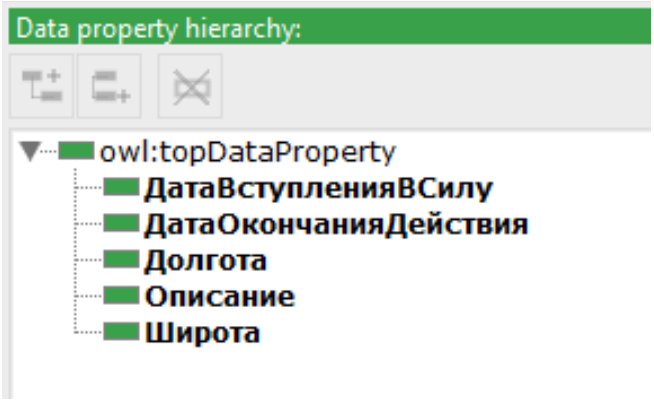

Fig. 4. Ontology data relationships

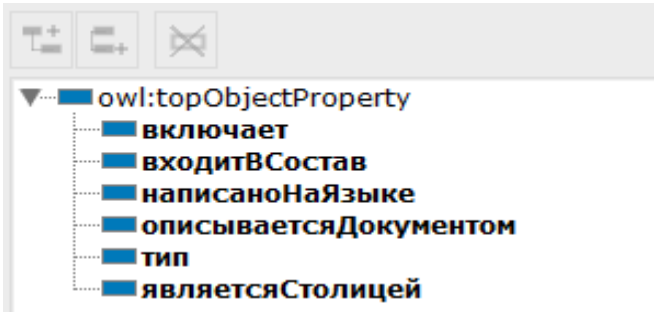

Fig. 5. Relationships of ontology objects

As a resource for filling, we used the Directory of Geographical Names of Kazakhstan [12]. The pattern matching method was applied to solve the problem of disclosure of abbreviations. And to do this, discovered during tokenization reduction was compared with the words given in the dictionary of abbreviations with templates based on regular expressions. The dictionary was used to expand the contraction of the input text.

V. Named Entity Unification Based on Geographic Thesauri

The task of unifying named entities based on geographic thesauri can be formulated as following:

- Let an ontology / vocabulary be given containing a set of entities E and a text containing a set of mentions to entities M,
- The task of linking entities is mapping each named mention of entities m in a given text to the corresponding entity e in a given ontology, here $m \in M$ and $e \in E$.

This task can also be named as entity normalization, entity grounding, or entity categorization. Let us explain it with the following example. Let there be a dictionary entity e1, a concept that designates the city of Nur-Sultan. Also let there be mentions to entities m1: Nur-Sultan, m2: Astana, m3: capital, etc. For an intelligent system that provides decision support using named entities, it is critical that all these mentions are reduced to a single concept.

The connection between mentioning of an entity and the concept of ontology can be found due to the lexical similarity between them. For example, the relationship between the mentions of the entities Tulebaika and Tulebaev street can be determined through the Levenshtein distance or lexical-morphological templates. However, lexical similarity may not always exist between references to entities and names or synonyms of concepts.

For instance, there is no lexical similarity between the mentions of Nur-Sultan and Astana, which requires the use of either a model to determine semantic similarity, or the use of ontological resources: address classifiers, GIS directory, geographic thesauri, etc.

Early systems attempted to link entity mentions to knowledge base entities using dictionary lookup and string matching algorithms [13,14]. In [15,16] manual rules were used to measure the morphological similarity between entity references and ontological concepts, while in [17], the patterns of entity variation were automatically investigated. Machine learning approaches examine the similarities between entity

references and ontology concepts based on labeled training data [18].

This work uses the approach proposed by the authors of [19]. For a set of documents with annotated named entities and the corresponding ontology, the normalization task is performed in two stages:

1) At the first stage, semantically similar concepts of ontology are generated as candidates.

2) In the second stage, candidates are re-ranked according to syntactically weighted semantic similarities.

The proposed approach is based on the assumption that semantically similar words have similar vector spaces. Basically, the semantic similarity of mentions of named entities and terms of ontology concepts is calculated. The most similar ontology concept can be assigned as a normalized concept to refer to a named entity (Fig. 6). In order to calculate semantic similarity, each word is represented in the vector space as a real vector using the publicly available pre-trained word embedding model Word2Vec[20].

The model was trained using word vectors, inducing from a large collection of geographical texts by the Word2Vec tool [20]. The trained model was applied to each word to obtain the corresponding word vector. The vectors of the ontology concepts and the named entities mentioned were calculated similarly. For named entities and concepts consisting of several words, vector representations were calculated by averaging the vectors of the words included in them.

For example, the named entity "Almaty Palace of Sports" was extracted from the document. As this entity consists of several words, the calculation of the vector representation is as follows. A vector from the pre-trained Word2Vec model is represented as a token. The vectors of all tokens are summed up, and the resulting sum is divided by the number of tokens, in this case three. Thus, a normalized vector of real numbers of a composite nominal entity is obtained. To compare with the concept from the ontology, the vector representation calculated in a similar way, the cosine measure of proximity is used.

As a result of the experiments, we normalized references to named objects based on 30 documents. An example of normalization is shown in TABLE 2.

TABLE 2. EXAMPLE OF NORMALIZATION

The name of the general (generic) concept	The name of the mentioned entity reduced to a concept	Method used for normalization
Palace of Sports and Culture named after Baluan Sholak	Baluan Sholak Palace	Lexical similarity based on Levenshtein distance
	Baluan Sholak	Lexical similarity based on Levenshtein distance
	Almaty Palace of Sports	Semantic similarity based on Word2Vec

A. Conclusion

In the course of work on the project, a large amount of work of both scientific and technical nature was done. The main results of the work:

– the architecture of the knowledge extraction system was created;
– methods of distributed high-performance collection, storage and preprocessing of data have been developed;
– a model of information extraction based on ontologies and machine learning methods has been created;
– a decision-making model was developed using decision trees;
– mechanisms for integrating storage sources of the extracted data with data processing and analysis tools based on the modified MapReduce system and Spark technology have been developed;
– a prototype of a recommendation service was developed based on the obtained knowledge base;
– a web application for visualizing emergencies has been developed. According to the conclusions [21] of the work, for a web application, actual security threats should be considered as during development and production. For all, the potential security risks identified during the scenario and mitigation opportunities during the analysis phase must be

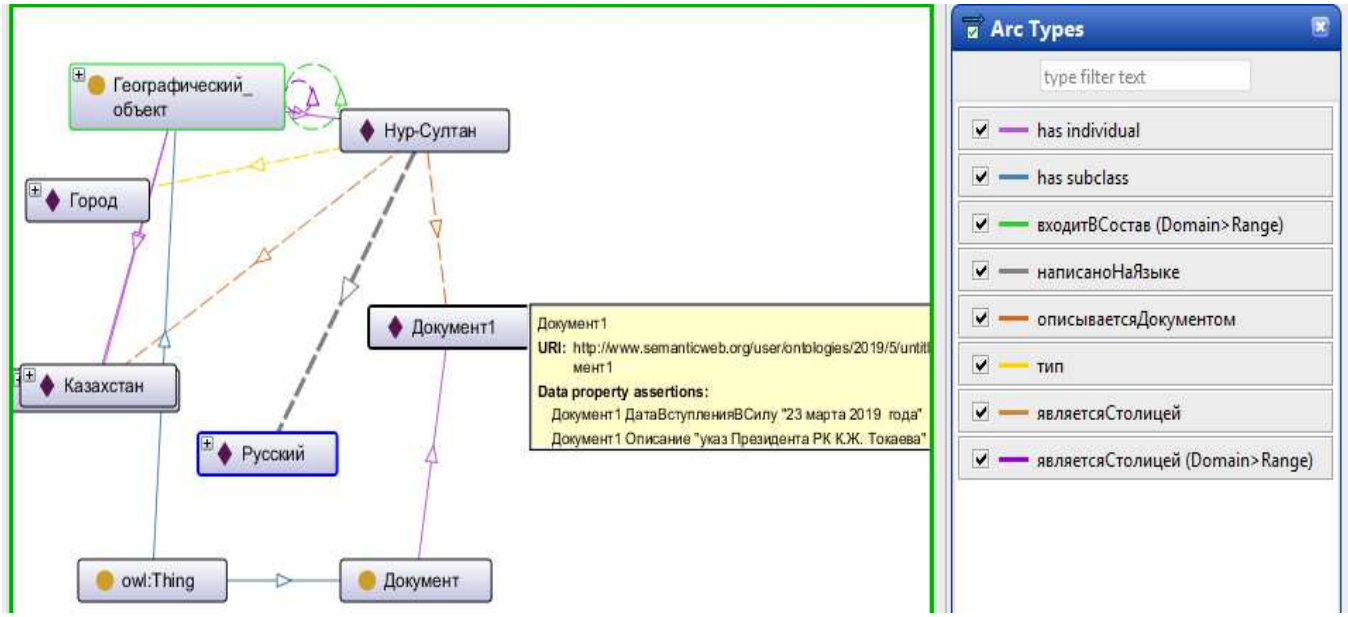

Fig. 6. Example of relations of the "Nur-Sultan" object

agreed. Nevertheless, special attention should as describe in [21] also be paid to:

• Authorization mechanisms
• Usage of digital certificates
• Encryption of transport protocols
• Encryption of data
• Usage of existent Virtual Private Network (VPN) infrastructure of involved parties
• Hacking defense complexes including IP blocking and DoS attack defenses.

The solution of the problem of named entities unification based on geographical ontologies is described in detail in this article. Information extraction models based on ontologies and machine learning methods were created.

Efficiency: development of a system for extracting knowledge from heterogeneous open sources, which provides support for decision making in any problem areas.
Applications: high-performance distributed computing, semi-structured data processing, intelligent decision support systems.

REFERENCES

[1] Zabasta, A., Kondratjevs, K., Peksa, J., Kunicina, N. MQTT enabled service broker for implementation arrowhead core systems for automation of control of utility' systems, Proceedings of the 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering, AIEEE 2017, 2018-January, pp. 1–6

[2] https://www.geonames.org [Accessed in May 2020]

[3] http://ruscorpora.ru/new/instruction-syntax.html (Accessed in May 2020)

[4] https: //www.nltk.org (Accessed in May 2020)

[5] M.Mansurova, V.Barakhnin, I.Pastushkov, E.Khibatkhanuly, M.Soltangeldinova Extraction of geographical names from semi-structured texts using machine learning algorithms // Proceedings of the II International Conference "Informatics and Applied Mathematics" ... Almaty, September 27-30, 2017 -C. 114-126

[6] Pascanu R., Mikolov T., Bengio Y. On the difficulty of training recurrent neural networks. 2012.

[7] K. Greff, R. K. Srivastava, J. Koutník, Bas R. Steunebrink, J. Schmidhuber. LSTM: A Search Space Odyssey. 2017.

[8] A. Nugumanova , Enrichment of the Bag-of-words model with semantic links to improve the quality of classification of domain texts // Software products and systems. - 2016. - No. 2 (114)

[9] Wei Y., Wei J., Xu H. Context vector model for document representation: a computational study //National CCF Conference on Natural Language Processing and Chinese Computing. – Springer International Publishing, 2015. – C. 194-206

[10] Chiu J.P.C., Nichols E. Named Entity Recognition with Bidirectional LSTM-CNNs. 2016

[11] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith.Transition-based dependency parsing withstack long short-term memory. // In Proceedings of ACL-2015 (Volume 1: Long Papers), Beijing, China, July. 2015. -P. 334–343.

[12] M. Lazarev's manual for the analysis of collection labels. URL: https://www.zin.ru/Animalia/Coleoptera/rus/kazgeonm.htm

[13] Fluck J., Mevissen H.T., Dach H., Oster M., Hofmann-Apitius M. Prominer: extraction of human gene and protein names using regularly updated dictionaries. // Proceedings of the Second BioCreAtIvE Challenge Evaluation Workshop. Madrid: Centro Nacional de Investigaciones Oncologicas, CNIO. 2007. -P. 149–51.

[14] Morgan AA, Lu Z, Wang X, Cohen AM, Fluck J, Ruch P, Divoli A, Fundel K, Leaman R, Hakenberg J, et al. Overview of biocreative ii gene normalization. // Genome Biol. 2008. 9(2):3

[15] Karadeniz İ, Özgür A. Detection and categorization of bacteria habitats using shallow linguistic analysis. // BMC Bioinformatics. 2015. 16(10).

[16] D'Souza J, Ng V. Sieve-based entity linking for the biomedical domain. // ACL (2). Beijing: Association for Computational Linguistics: 2015. 2003. -P. 297–302.

[17] Ghiasvand O, Kate RJ. Uwm: Disorder mention extraction from clinical text using crfs and normalization using learned edit distance patterns. // SemEval@ COLING. Dublin: 2014. -P. 828–32.

[18] Leaman R., Islamaj D. R., Lu Z. Dnorm: disease name normalization with pairwise learning to rank. // Bioinformatics. 2013. 29(22):2909–17

[19] Karadeniz I., Özgür A. Linking entities through an ontology using word embeddings and syntactic re-ranking //BMC bioinformatics. – 2019. – T. 20. – №. 1. – C. 156.

[20] Chiu B, Crichton G, Korhonen A, Pyysalo S. How to train good word embeddings for biomedical nlp. // Proc BioNLP16. 2016. 1:166–174.

[21] Dorogovs, P., Romanovs, A. Overview of government e-service security challenges. Advances in Information, Electronic and Electrical Engineering, AIEEE 2015 - Proceedings of the 2015 IEEE 3rd Workshop, 2015, 7367316