Named Entity Recognition in Tatar: Corpus-Based Algorithm

Olga Nevzorova[1][0000-0001-8116-9446], Damir Mukhamedshin[1][0000-0003-0078-9198], and Alfiya Galieva[1]

[1] The Tatarstan Academy of Sciences, Kazan, Russia
{onevzoro,damirmuh,amgalieva}@gmail.com

Abstract. Named entities recognition is one of the urgent tasks in the researches of language using electronic language corpuses. This article discusses the main methods for solving this problem, including algorithms based on various machine learning models, regular expressions and dictionaries. Also in the article, the authors proposed their own algorithm, which allows named entities recognition on the basis of search queries using direct and reverse search. The results of the algorithm, presented in the article, suggest what additional functions are necessary to achieve the best results. The proposed algorithm is used in the "Tugan Tel" corpus management system and can be used both with the electronic corpus of the Tatar language and with corpuses of other languages.

Keywords: Named entity recognition, NER, Corpus management system, Text mining.

1 Introduction

Electronic language corpuses are the basis for extensive research related to language research. Corpus management systems help solve a number of linguistic problems, such as direct search of word forms, lemmas, reverse search by morphological properties, selection of contexts, n-grams for various search queries. These simple queries are supported by most corpus management systems.

One of the difficult tasks of searching in corpus data is named entities recognition. This problem is solved by dozens of researchers, often getting good results. Most existing solutions, some of which are described in Section 2 of this article, work with English, Spanish, Dutch, German using various NLP methods, regular expressions, dictionaries, etc. as the basis. In Section 4 of this article, the authors considered one of the possible algorithms for named entities recognition, which can be used both with the electronic corpus of the Tatar language and with electronic corpuses of other languages. This algorithm is implemented in one of the modules of the "Tugan Tel" corpus management system. The authors also conducted a series of experiments, the results of which are shown in Section 4.2 of this article.

2 “Tugan Tel” Corpus Management System

The Tatar corpus management system (www.corpus.antat.ru) is developed at Institute of Applied Semiotics of the Tatarstan Academy of Sciences. The main functions of the corpus management system are searching for lexical units, making morphological and lexical searches, searching for syntactic units, n-gram searching based on grammar and others. The core of the system is the semantic model of data representation. The search is performed using common open source tools. We use MariaDB database management system and Redis data store [1]. Our purpose is to design the corpus management system for supporting electronic corpora of Turkic languages. This line of research is developing very rapidly.

Among well-known electronic corpora projects for Turkic languages are the corpora of Turkish and Uyghur [2], Bashkir, Khakass, Kazakh (http://til.gov.kz), and Tuvan languages. “Tugan Tel” Tatar national corpus is a linguistic resource of modern literary Tatar. It comprises more than 100 million word forms, at the rate of November 2016. The corpus contains texts of various genres: fiction, media texts, official documents, textbooks, scientific papers etc. Each of the documents has a meta description [3]: author, title, publishing details, date of creation, genre etc. Texts included in the corpus are provided with morphological markup, i.e. information about part of speech and grammatical properties of the word form [4]. The morphological markup is carried out automatically on the basis of the module of two-tier morphological analysis of the Tatar language with the help of PC-KIMMO software tool.

3 Related Works

3.1 LingPipe

One of the related works is LingPipe [5], which is a collection of Java libraries developed by Alias-I. LingPipe allows to classify named entities in English: person, organization, place. It supports the use of other language packages for classification. LingPipe also supports additional features such as orthographic correction and English text classification. This software is distributed free of charge for research purposes.

3.2 Annie

Another similar work is Annie [6]. This is a named entity extraction module embedded into the GATE framework. Annie is open source and is developed under the GNU license developed at Sheffield University. Annie implements various functions necessary for extracting named entities: tokenizer, sentence separator, POS tagging, resolution with a link, place name directories, etc.

3.3 Afner

Afner [7] is an open source NERC tool licensed under the GNU license, developed in C++ at Macquarie University. It is used as part of a question and answer service that focuses on maximizing responsiveness to user questions. At the same time Afner can be used separately from the service. Afner uses lists, regular expressions, and supervised learning models. It allows one to extract names of persons, organizations, locations, monetary values and dates from English texts.

3.4 Knowledge-based systems

Knowledge-based NER systems use lexical resources and domain-related knowledge without requiring training with annotated data. Such systems show good results when the lexical resources are complete, whereas they do not work, for example, with the examples from drug_n class in the DrugNER [8] data set, since they are not defined in the DrugBank dictionaries. Despite their high accuracy, these systems show low recall due to specific rules of the language and domain and incomplete dictionaries. Another disadvantage of knowledge-based NER systems is the need for experts to participate in the development and maintenance of knowledge resources.

3.5 Unsupervised and bootstrapped systems

Early systems did not require significant data for training. Collins and Singer (1999) [9] used only labeled seeds and 7 functions for classifying and extracting named entities: orthography (for example, capitalization), entity context, words that occurred in named entities, etc. To improve the recall of NER systems, Etzioni et al. (2005) [10] proposed an unsupervised system using 8 generic pattern extractors for open web texts, for example, *NP is <class1>*, *NP1 such as NPList2*. In 2006, Nadeau et al. suggested using an unsupervised system to create a directory of named entities and resolve the ambiguity of named entities basing on the work of Etzioni et al. (2005) [10] and Collins and Springer (1999) [9]. This system combined the extracted list of named entities with generally accessible directory of named entities and achieved F-scores of 88%, 61% and 59% on MUC-7 [11] for named entities of classes of locations, persons and organizations, respectively.

Zhang and Elhadad (2013) [12] in an unsupervised NER system for biological and medical data used surface syntactic knowledge base and inverse document frequency (IDF). This system reached 53.8% and 69.5%, respectively. Their model uses seeds to find text with possible content of named entities, identifies phrases with nouns and filters phrases with a low IDF value. The filtered list is submitted to the classifier for predicting the tags of named entities.

3.6 Feature-engineered supervised systems

Supervised machine learning models learn to make predictions by training on example inputs and their expected outputs, and can be used to replace humanly established rules. Hidden Markov Models (HMM), Support Vector Machines (SVM), Conditional Random Fields (CRF), and decision trees were common machine learning systems for NER.

The results of research using various machine learning models from various authors are presented in Table 1.

Table 1. Various machine learning models results.

Author(s)	Machine learning model	Additions	Results
Zhou and Su (2002) [13]	HMM	Included 11 orthographic features, a list of trigger words for named entities, and a list of words from various gazetteers.	F-scores of 96.6% and 94.1% on MUC-6 and MUC-7 data, respectively.
Malouf (2002) [14]	HMM and Maximum Entropy (ME)	Included capitalization; considered whether the word went first in the sentence, whether the word had appeared before with a known last name, and 13281 first names collected from various dictionaries.	F-scores of 73.66% and 68.08% on Spanish and Dutch CoNLL 2002 datasets, respectively.
Carreras et al. (2002) [15]	Binary AdaBoost classifiers	Included capitalization, trigger words, previous tag prediction, bag of words, gazetteers.	F-scores of 81.39% and 77.05% on Spanish and Dutch CoNLL 2002 datasets, respectively.
Li et al. (2005) [16]	SVM	Experimented with multiple window sizes, features (orthographic, prefixes suffixes, labels, etc.) from neighboring words, weighting neighboring word features according to their position, and class weights to balance positive and negative classes.	F-score of 88.3% on the English CoNLL 2003 data.
Ando and Zhang (2005) [17]	Structural learning [17]	The best classifier for each auxiliary task was selected based on its confidence.	F-scores of 89.31% and 75.27% on English and German, respectively.
Agerri and Rigau (2016) [18]	Semi-supervised system	Included orthography, character of n-grams, lexicons, prefixes, suffixes, bigrams, trigrams, and unsupervised cluster features from the Brown corpus, Clark corpus and k-means clustering of open text using word embeddings.	F-scores of 84.16%, 85.04%, 91.36%, 76.42% on Spanish, Dutch, English, and German CoNLL, respectively.

4 Extracting named entities

Extracting named entities from corpus data allows, on the one hand, to directly retrieve the required data by query, and on the other hand, to test the corpus for containing particular information and to replenish it with documents that include the missing data. The algorithm of extraction of named entities proposed in this paper enables to obtain semantic samples for corpora that do not have semantic data markup. On the other hand, the algorithm has no restriction on semantic types of extracted data, i.e. the semantic type is defined by the keyword in the query.

4.1 Describing algorithm of extracting named entities

The algorithm for extracting named entities is based on the idea of comparing n-grams. The comparison is made within the entire corpus volume, thereby increasing the accuracy of the results.

The extraction process is iterative, the threshold number of iterations specified by the user. The first step presents sampling by the initial search query. The initial search query may be a query on the word form, lemma or phrase, or a search by morphological parameters. A list of bigrams and their frequency is collected across the sample. The bigrams which contain the results are advanced one position to the left or right (set by the user). The resulting list is sorted by frequency of bigrams in order from largest to smallest, to be cut to a predetermined covering index (for example, 95% of all results, this rate being set by the user). This result is used in the second iteration of the algorithm. Each bigram is searched for in the mode of phrasal search in the corpus. Search results are involved in composing a list of trigrams which are advanced one position to the left or right, and their frequency. The resulting list of trigrams is also sorted by frequency in order from largest to smallest, and is cut to a predetermined covering index.

The third and subsequent iterations (until the threshold number of iterations is reached or no match is found as a result of iterating) use the list of n-grams received from the previous iteration. The corpus is searched for each n-gram in the phrasal search mode, and a list of (n + 1)-grams is made up. The resulting list is then cut to a predetermined covering index and compared with the list of n-grams derived from the previous iteration. The comparison accuracy P is set by the user as a percentage. If n-gram frequency is less than P from the quantity of the found (n + 1)-gram, then the n-gram is considered the found named entity, otherwise the extraction proceeds. Thus, the final result will represent a list of the most stable n-grams of different lengths, including search results by the initial search query.

A request to retrieve named entities is an extension of a Q-tuple presented in (1). In addition to the search query, there are added components defining the threshold number of iterations to the left (L) and right (R), the covering index (C), and the accuracy of matching (P). A search example is presented in (1).

$$Q = (Q_1, Q_2, L, R, C, P) \tag{1}$$

4.2 Experiments

Extracting named entities using the algorithm proposed by the authors requires an initial search query which should contain an indicator of a particular named entity. This indicator allows classifying named entities, therefore, the authors chose a set of classes schema.org as the basis for choosing the indicators. From this set of classes, the authors selected the following classes for searching for named entities in the Tatar language corpus: books, restaurants, films, magazines, companies, airports, corporations, languages, technical schools, universities, schools, shops, museums, and hospitals. Ministries and street names have also been added to this list. Below are some of the results of the experiments conducted by the authors.

Names of ministries

As part of the task of enhancing named entity search a number of experiments have been carried out. One of the most revealing of them was search for names of ministries. The initial search query for the experiment was (2).

$$Q = ((\text{wordform}, \textit{ministrlygy}, \text{“”}, \text{right}, 1, 10, \text{exact}), 7, 0, 95, 80) \qquad (2)$$

The result of this query was a list of 50 n-grams containing word form "*ministrlygy*" in the last position. The reference list of names of ministries presented on the Republic of Tatarstan government website [http://prav.tatarstan.ru/tat/ministries.htm] contains 17 items. 12 of 17 items were found in the corpus by means of the algorithm, so the results overlap is 70.6%. 5 items were not found in the corpus for the reasons described in Table 2. The remaining 33 n-grams are different spelling variants of names of ministries.

Table 2. List of unfound names of ministries.

Name	Reason
Urman huҗalygy ministrlygy (Tat) – ministry of forestry	Overlap of the sequence of word forms with the sequence in another name «huҗalygy ministrlygy» (Tat) – ministry of property and «Transport həm yul huҗalygy ministrlygy» (Tat) – ministry of transport and road management
Yashlər eshləre həm sport ministrlygy (Tat) – ministry of youth and sport	Corpus meanings not corresponding to the official name
Transport həm yul huҗalygy ministrlygy (Tat) – ministry of transport and road management	Overlap of the sequence of word forms with the sequence in another name «huҗalygy ministrlygy» (Tat) – ministry of property and «Urman huҗalygy ministrlygy» (Tat) – ministry of forestry
Hezmət, halykny el belən təemin itү həm social yaklau ministrlygy (Tat) - ministry of labour, employment and social	Corpus meanings not corresponding to the official name

protection	
Ecologia həm tabigy baylyklar ministrlygy (Tat) – ministry of ecology and natural resources	Corpus meanings not corresponding to the official name

Names of streets

Another experiment was concerned with street names search. The search query for this experiment is (3).

$$Q = ((\text{wordform}, \textit{uramy}, \text{“”}, \text{right}, 1, 10, \text{exact}), 7, 0, 95, 80) \quad (3)$$

The result of this query was a list of 600 n-grams containing word form "*uramy*" in the last position. We obtained the following results after manual data evaluation: 432 (72%) n-grams are street names, 72 (12%) n-grams are also street names, but require special character filtering, 96 (16%) n-grams are not street names for various reasons (for example, any sentences containing the word “uramy”; postal addresses and others).

Names of languages

In the next experiment, the authors tried to extract names of languages. The search query for this experiment is presented in (4).

$$Q = ((\text{wordform}, \textit{tel}, \text{“POSS\_3SG,SG”}, \text{right}, 1, 10, \text{exact}), 7, 0, 95, 80) \quad (4)$$

After executing this query, 2310 n-grams were obtained, containing “*tel*” lemma with the morphological properties POSS_3SG and SG in the last position. An estimation of part of the results (a list of 471 n-grams) by an expert showed that in 53.5% of cases (252) n-grams were correct language names. Analysis of the list of n-grams which were incorrectly defined by the algorithm as a name of a language, made it possible to determine additional filtering rules to improve the accuracy of the algorithm. On the basis of the data obtained, the spreading of language names in the corpus of the Tatar language was also constructed (Fig. 1).

Names of restaurants

Another experiment is related to search for names of restaurants. The search query for this experiment is presented in (5).

$$Q = ((\text{wordform}, \textit{restoran}, \text{“POSS\_3SG,SG”}, \text{right}, 1, 10, \text{exact}), 7, 0, 95, 80) \quad (5)$$

The result of this query was a list of 285 n-grams containing “*restoran*” lemma with the morphological properties POSS_3SG and SG in the last position, which in total were found 359 times in the corpus. In this case, in addition to names of restaurants, names of sub-classes of restaurants by their geographical location or national cuisines were obtained.

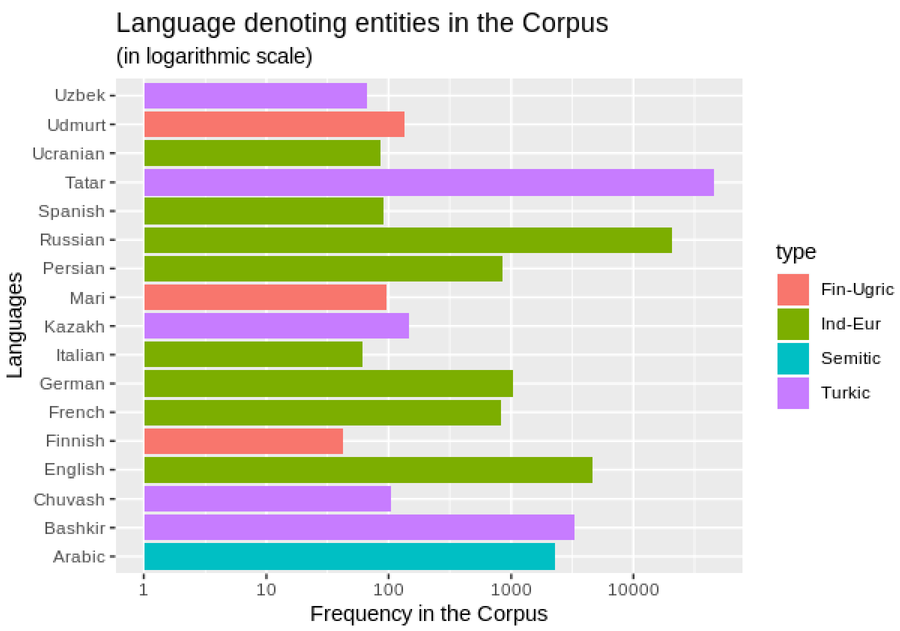

Fig. 1. Language denoting entities in the Corpus.

Thus, 107 (37.68%) found n-grams were correct names of restaurants, their total frequency being 140 (39%). 37 (13.03%) n-grams were the names of subclasses of restaurants, their total frequency being 47 (13.09%). 52 (18.31%) n-grams contained names of restaurants, but they require cleaning from unnecessary parts, while the frequency of the n-grams in the corpus is 2 or less, the total frequency is 54 (15.04%). 45 (15.85%) n-grams contained names of subclasses of restaurants, but they require cleaning from unnecessary parts, while the frequency of n-grams in the corpus is 2 or less, the total frequency is 48 (13.37%). 43 (15.14%) n-grams were not names of restaurants, their total frequency was 65 (18.11%). The list of incorrectly defined n-grams can be reduced by applying additional filtering rules.

Names of corporations

The next experiment was the search for names of corporations. The search query for this experiment is presented in (6).

Q = ((wordform, *korporaciya*, “POSS_3SG,SG”, right, 1, 10, exact), 7, 0, 95, 80)   (6)

As a result of this search query was obtained a list of 138 n-grams containing lemma “*korporaciya*” with morphological properties POSS_3SG and SG in the last position, which were found in the corpus 606 times. Among them, when checked by an expert, 63 (45.65%) n-grams were found, which were correct names of corporations, their total frequency being 178 (29.37%). 27 (19.57%) n-grams contained names of corporations, but require additional cleaning; the total frequency of these n-grams was

29 (4.79%). Among the results, 15 (10.87%) n-grams were singled out, which were non-full names of corporations, their total frequency being 58 (9.57%). 30 (21.74%) n-grams were names of subclasses of corporations by industry, geography, government participation; such n-grams were found in the corpus 336 times (55.45%). 3 (2.17%) n-grams were not names of corporations, their total frequency being 5 (0.83%).

Comparison of results

For different classes of named entities, the algorithm shows different results. The results presented in this article are shown in Table 3.

Table 3. Experiments results.

Class of named entity	Correct	Require filtering	Require expansion	Correct names of subclasses	Names of subclasses that require filtering	Incorrect	Total
Names of ministries	100%	0%	0%	0%	0%	0%	50
Street names	72%	12%	0%	0%	0%	16%	600
Language names	53.5%	0%	0%	0%	0%	46.5%	471 (2310)
Restaurant names	37.7%	18.3%	0%	13%	15.9%	15.1%	285
Corporation names	45.7%	19.6%	10.9%	21.7%	0%	2.2%	138

4.3 Temporal and qualitative indicators of implementing a query for extracting named entities

The experiments showed that the time of implementing a query for extracting named entities depends on the number of found items and bigrams by the initial search query, and on indexes of covering and the accuracy of comparison. All the experiments were executed on machine with following characteristics: 4 core Intel Core i7 2600 (2,6GHz), 16GB RAM (4x4GB, 1333Hz), SSD 120GB, HDD 3TB (3x1TB, RAID 0). On the test machine Ubuntu Server 14.04 LTS was running. Table 4 shows the timing indicators of search implementation. Algorithm tests revealed dependence of the quality of the results on the number of results found in the first step of the algorithm. This is due to the fact that a smaller number of results increase the actual data coverage and the data which the algorithm works with may initially include particular cases. More results in the first step suggest that at the first cutting of the bigram list, only those will remain that will be included in the final list of the extracted named entities. Thus it is only needed to find the left or the right border for this list.

Table 4. Temporal indexes of implementing searches for extraction of named entities.

Search query	Quantity of found items	Quantity of found bi-grams	Time elapsed
Q = ((wordform, ministrlygy, “”, right, 1, 10, exact), 7, 0, 97, 80)	27746	68	127.37 sec.
Q = ((wordform, uramy, “”, right, 1, 10, exact), 3, 0, 95, 80)	9592	600	848.07 sec.

5 Conclusion

The algorithm for named entity recognition proposed by the authors in this article shows different results, depending on the type of named entities. The presented results demonstrate correctness of recognition from 37.7% to 100%.

In addition to the main task of named entity recognition, the algorithm is applicable for solving the problem of recognition of names of subclasses of named entities. This feature can be applied to solve additional problems, such as text classification, definition of the subject of texts and other text mining tasks.

Analysis of the results obtained during the experiments show that to improve the accuracy and correctness of the algorithm, its fine tuning, building extended dictionaries for named entity recognition, and additional post-processing of results are necessary.

References

1. Nevzorova O., Mukhamedshin D., Gataullin R. Developing Corpus Management System: Architecture of System and Database. Proceedings of the 2017 International Conference on Information and Knowledge Engineering. CSREA Press, United States of America, pp. 108-112 (2017).
2. Aibaidulla Y., Lua K.T. The development of tagged Uyghur corpus. Proceedings of PACLIC17, pp. 1–3 (2003).
3. Nevzorova, O., Mukhamedshin, D., Kurmanbakiev, M. Semantic aspects of metadata representation in corpus manager system. Open Semantic Technologies for Intelligent Systems (OSTIS-2016), pp. 371–376 (2016).
4. Suleymanov, D., Nevzorova, O., Gatiatullin, A., Gilmullin, R., Hakimov, B. National corpus of the Tatar language “Tugan Tel”: grammatical annotation and implementation. Proc. Soc. Behav. Sci. 95, pp. 68–74 (2013).
5. Baldwin B., Carpenter B. LingPipe, http://alias-i.com/lingpipe, last accessed 2018/10/12.
6. Bontcheva K., Dimitrov M., Maynard D., Tablan V., Cunningham H. Shallow methods for named entity coreference resolution. Chaınes de références et résolveurs d’anaphores, workshop TALN. (2002)

7. Zaanen M., Molla D. A named entity recogniser for question answering. Proceedings PACLING (2007)
8. Segura Bedmar I., Mart´ınez P., Herrero Zazo M. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics (2013).
9. Collins M., Singer Y. Unsupervised models for named entity classification. 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999).
10. Etzioni O., Cafarella M., Downey D., Popescu A.-M., Shaked T., Soderland S., Weld D., Yates A. Unsupervised named-entity extraction from the web: An experimental study. Artificial intelligence, 165(1), pp. 91–134 (2005).
11. Chinchor N., Robinson P. Muc-7 named entity task definition. In Proceedings of the 7th Conference on Message Understanding, 29 (1997).
12. Pradhan S., Moschitti A., Xue N., Tou Ng H., Bjorkelund A., Uryupina O., Zhang Y., Zhong Z. Towards robust linguistic analysis using ontonotes. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning, pp. 143–152 (2013).
13. Zhou G., Su J. Named entity recognition using an hmm-based chunk tagger. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics. Pp. 473–480 (2002).
14. Malouf R. Markov models for language-independent named entity recognition. Proceedings of the 6th conference on natural language learning, 31 (2002).
15. Carreras X., Marquez L., Padro L. 2002. Named entity extraction using adaboost. Proceedings of the 6th conference on natural language learning, 31 (2002).
16. Li Y., Bontcheva K., Cunningham H. Svm based learning system for information extraction. Deterministic and statistical methods in machine learning. Springer. Pp. 319–339 (2005).
17. Ando R.K., Zhang T. A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research, 6 (Nov), pp. 1817–1853. (2005).
18. Agerri R., Rigau G. Robust multilingual named entity recognition with shallow semi-supervised features. Artificial Intelligence, 238, pp. 63–82 (2016).