DOI 10.15276/imms.v10.no1-2.23
УДК 004.912

Informatics and Mathematical Methods in Simulation
Vol. 10 (2020), No. 1-2, pp. 23-30

DEVELOPMENT OF SOFTWARE FOR THE CREATION OF THE CORPUS OF THE UKRAINIAN LANGUAGE AND ITS USE

O.I. Kerpel, V.G. Penko

Odessa I.I.Mechnikov National University,
2, Dvoryanskaya St., Odessa, 65026, Ukraine, e-mail: alexeykerpel15@gmail.com, vpenko@onu.edu.ua

The relevance of the work lies in the need to analyze Ukrainian texts in order to study the Ukrainian language and the lack of appropriate information and software resources. The object of research is the variety of tools for the task of POS tagging of Ukrainian texts. The subject of the research is the process of developing software for creating corpora of Ukrainian texts, in particular, POS tagging algorithms. The purpose of the work is to research and develop software for creating an annotated corpus of the Ukrainian language. To achieve this goal, following tasks were solved: analysis of the subject area; selection of suitable software tools; creating a training set; system implementation; system training; system testing. An approach to the generation of a training sample was designed and implemented. The approach is based on the use of an already implemented tagger of the Russian language and the similarity of the morphological structure of words in the Russian and Ukrainian languages. The search for an effective combination of the feature space and the training algorithm was made. The most successful machine learning models for this task turned out to be stochastic gradient descent and decision trees. K-fold cross-validation was used to achieve an acceptable level of generalization. As a result of this work, a trained tagger of the Ukrainian language was obtained, which, for a sample of literary Ukrainian texts, provides the quality of classification at the level of 0.892 according to the weighted F-measure. A distributed application with a client-server architecture has been implemented, which allows clients to tag their own texts. With the regular use of this application by linguistic experts, the training set used can be improved, which will lead to higher rates of classification carried out by the tagger.

Keywords: POS tagging, text corpus, machine learning algorithms, feature space.

Introduction

At the moment, processing of the natural texts is becoming more and more demanded in different contexts. Among them are speech recognition systems (Cortana, Siri), search engines (Google, Yandex, Yahoo), etc. Also, computer processing of natural texts is necessary to detect plagiarism which is present in the works of various authors by identifying certain patterns in the texts of different genres. Word processing can be useful in tasks such as organizing search in search engines, speech recognition, automatic detection of topics on web pages, translation of texts, and in many similar tasks that require advanced word processing.

Within text analysis research area, text corpora (plural form of corpus notion) are commonly used. Corpora are collections of various texts that allow you to study the language from different angles and contain different types of markup (meta-data, such as morphological, semantic, syntactic, etc.). Until recently, the only possible way to compose such a corpus was to tag texts manually, but the development of information technologies opened up new opportunities by enabling markup process automation and significantly reducing (or completely eliminating) the need for human (expert) intervention in the word recognition area.

Unfortunately, for the Ukrainian language there is still lack of marked-up corpora and software resources sufficient for practical use, which would allow the natural text markup automation.

Up till now, there is only one corpus of the Ukrainian language developed by the KNU. T. Shevchenko. Despite the declared presence of part-of-speech markup, in fact, the search for parts of speech on the official website does not work [1], and the search is available exclusively in legislative texts. There is also a project according to which the development of the corpus of the Ukrainian language from NU Ostrozka Academy is underway. Work on the project started in 2010. However, there is essentially no progress in the development of the project [2].

The goal of this work is to create an automated system that allows you to generate text markup as a basis for creating corpora of Ukrainian texts in future.

Given the variety of tagging types, construction of universal tools of this kind is hardly possible. Due to this limitation, it was decided to focus on obtaining part-of-speech markup or POS tagging.

To achieve defined goal, following tasks were accomplished:

- analysis of the POS tags system used in the Ukrainian language;
- investigation of the possibility of building simple automatic taggers (usually based on regular expressions). Evaluation the effectiveness of such taggers for the Ukrainian language;
- investigation of the approaches to building an automated tagger based on machine learning methods in terms of their applicability to the Ukrainian language;
- implementation of the automated tagger and analysis of the effectiveness of its performance;
- implementation of the distributed system to provide access to the classification system based on the automated tagger.

Main Part

At the first stage, there is a need to clarify the system of the Ukrainian tags. The system of parts of speech of the Ukrainian language is generally accepted [3]. In the process of implementing the POS tagger, it was decided to use standard tag notation, shown on the table 1.

Table 1.

List of the Ukrainian POS-tags

Independent POS		Tag	Auxiliary POS	Tag
Іменник		S	сполучник	CONJ
Прикметник		A	прийменник	PR
Числівник		NUM	частка	PART
Займенник		A-PRO		
Дієслово		V		
Прислівник		ADV		

Simple taggers.POS-tagging is a well-known problem in the field of natural text processing problems. To date, a wide range of diverse approaches to solving this problem have been proposed [4]. There are two relatively simple possibilities for building taggers: based on dictionaries or regular expressions.

In the case of a dictionary usage, it will be necessary to create a dictionary consisting of all the words of the language, as well as all their forms, which will require a huge waste of resources and is an extremely difficult and time-consuming task. And still, there are many ambiguity cases.

A regex-based tagger is simpler, but has a disadvantage of being not suitable for all languages. Regular expression tags are very different depending on a languages. Building such taggers requires deep expertise in the grammar of a particular language. It has a fairly low tagging accuracy (even for English - about 85% on average versus machine learning-based taggers - 97%). To improve accuracy, it is necessary to continuously increase the number of regular expressions, which is highly complex and effort-consuming task, and it is also less effective than analogs based on machine learning. A regular expression tagger is barely suitable for tagging Slavic languages due to the high number of different cases and word forms, which, in its turn, leads to extremely low accuracy of tag recognition in the text.

In connection with the above, systems based on machine learning have been selected as the most effective means for solving the problem of POS-tagging of the Ukrainian language.

Training set. The absence of tagged Ukrainian texts is an obstacle to obtaining a training set within the framework of the problem being solved. To solve this problem in this work, it was decided to use the scheme involving the combination of Russian and Ukrainian languages, as languages belonging to the same Slavic group. One of the features of this similarity is the similar word order in the sentence. Due to this similarity, as well as the presence of the Russian language POS tagger in the NLTK library, it became possible to take Ukrainian texts, split them into words, translate them into Russian words and tag them as the Russian text, and then match the tags to the Ukrainian words as shown on the figure 1. Thus, it was possible to automatically obtain a sufficiently large and high-quality sample base which can be used as training set in machine learning experiments.

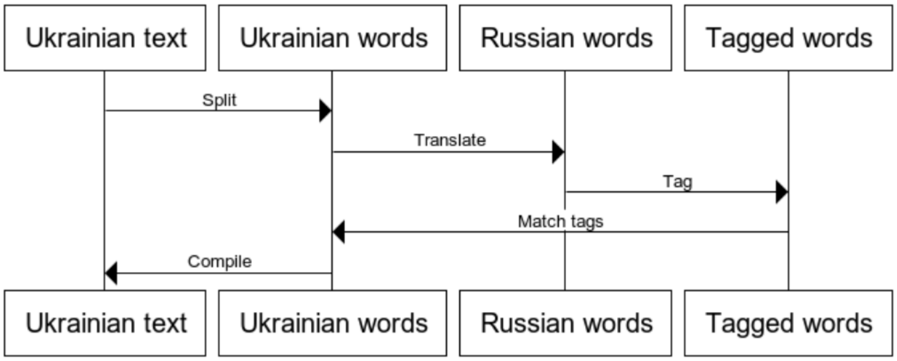


Fig. 1. Training set construction

The training set contains the fiction texts as shown on the table 2.

Table 2.

Training set texts

Texts (Ukrainian translation)	Words
Dead Souls, by Nikolai Gogol	3114
The Master and Margarita by Mikhail Bulgakov	2101
War and Peace by Leo Tolstoy	2283

Note that Ukrainian translations of famous literature works were chosen as the source texts. At this stage, relatively small fragments of these texts have been used.

Features. One of the most elaborate operations in creation of an effective automated tagger based on classifiers is the determination of a set of entity attributes. These attributes affect its compliance with certain classes. In the field of machine learning these entity attributes are called features. Determining the appropriate feature sets is critical. In this case, the most effective features can be present in the text in an implicit form. Research toward finding successful feature sets and their subsequent use represents an important subsection of machine learning, commonly referred to as feature engineering [5,6].

Due to the importance of identifying an effective set of features, a study was conducted to identify the optimal set of features. The task of identifying the best features is tightly associated with the task of determining the most appropriate learning algorithm. However, an exhaustive computational experiment in this two-dimensional space is extremely time-consuming. In this work, a simplified approach is used. The SGD classifier was chosen as a basic classifier to determine the effectiveness of certain features, shown in table 3.

The feature search strategy is based on the idea of revealing the most intuitive possible features and then checking their compatibility with each other.

Table 3.

Class features

Features	Meaning	Weighted F-score
word	The word itself	0.78
word + is_first	If the word is first in sentence	0.78
word + is_last	If the word is last in sentence	0.775
word + position	The position of the word in sentence	0.76
word + capitalized	If the word is capitalized	0.776
word + is_all_caps	If all the letters are uppercase	0.778
word + is_all_lower	If all the letters are lowercase	0.7785
word + prefix-1	First letter	0.72
word + prefix-2	First 2 letters	0.733
word + prefix-3	First 3 letters	0.77
word + suffix-1	Last letter	0.73
word + suffix-2	Last 2 letters	0.8
word + suffix-3	Last 3 letters	0.82
word + suffix-4	Last 4 letters	0.82
word + all suffixes	All suffixes	0.845
word + all prefixes	All prefixes	0.766
word + all suffixes and prefixes	All suffixes and prefixes	0.89
word + prev_word	The word before the current	0.76
word + next_word	The word after the current	0.75
word + has_hyphen	If the word has hyphen	0.78
word + is_numeric	If the word consists of digits	0.78
word + capitals_inside	If the word has capitals inside	0.78
All the above	All the features	0.89
All the above – capitalized	All the features except capitalized	0.892
All the above – capitalized – prev_word – next_word	All the features except capitalized, previous and next words	0.892

Machine learning algorithms. Another important aspect that affects the performance and efficiency of classification is the type of machine learning algorithms. Machine learning scientists have developed a huge number of these algorithms. Our practical choice was limited to a basic set of algorithms that are implemented in the popular scikit_learn programming tool [7].

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression.

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.

A multilayer perceptron (MLP) is a type of artificial neural network.

An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.

K-neighbours. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the highest number of representatives within the nearest neighbors of the point.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.

The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

The mentioned above classifiers were compared on the selected set of features on different training set range. To obtain better generalization, K-block cross validation was used [8]. The results are shown on figure 2.

As we can see on the figure. 2, the best classifiers, according to the weighted F-score is SGD classifier. So we can choose it for the following application implementation.

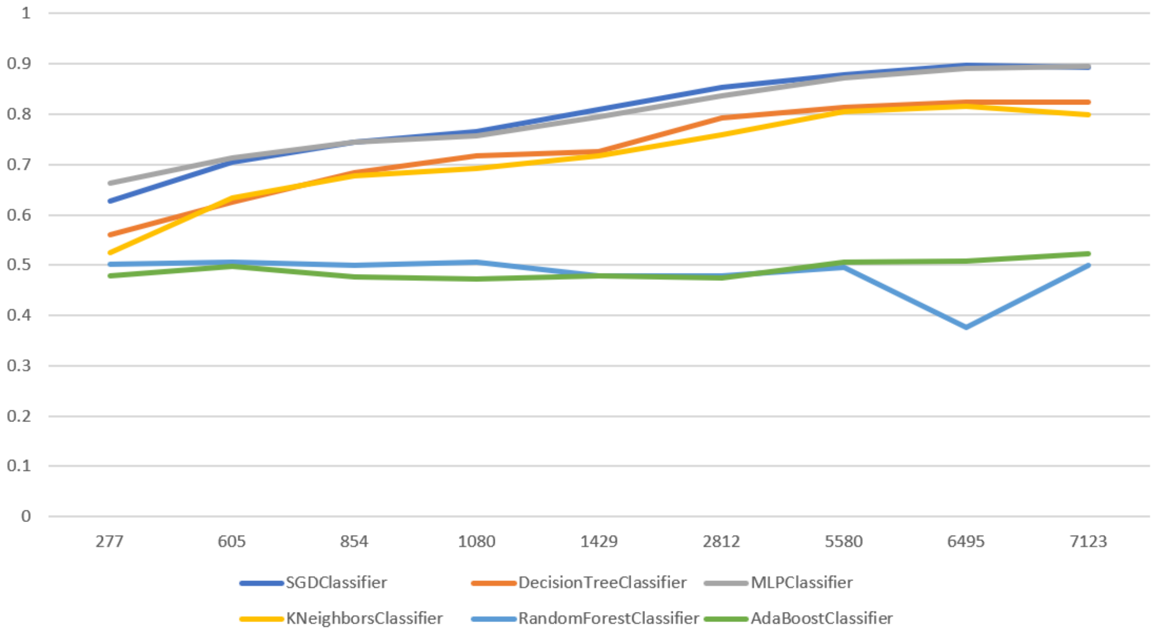
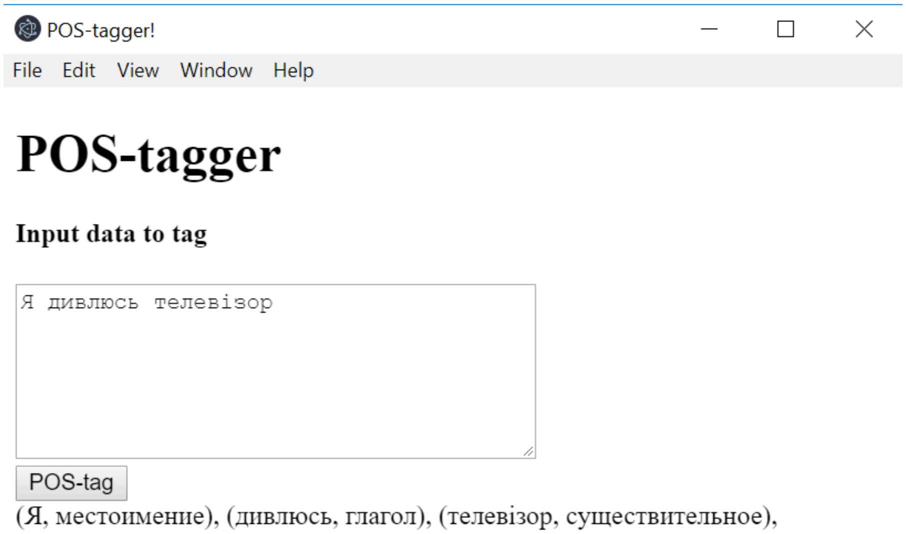


Fig. 2. Weighted F-score classifiers comparison

Client-server architecture. The task of the implementation of this work is to create the system that would provide accurate POS predictions for the user, but at the same time the goal of this work implies the facilitation for the linguists that would use it to create the Ukrainian text corpus.

It was decided to create the distributed system that would consist of the client and the server to provide the ability of using the tagger remotely. The Electron app was chosen as the client platform, as it can run on any operating system [9]. The server was written in Python same as the main learning application as it is efficient enough to handle the simple http requests.

As shown on the figure 3, the client is displaying the input field where the user can type any word combination or text he wants to tag and the "POS-tag" button that sends the request for tagging to the server. The server responses with the tagged words which get displayed in the application interface.



POS-tagger

Input data to tag

Я дивлюсь телевізор

(Я, местоимение), (дивлюсь, глагол), (телевізор, существительное),

Fig. 3. Application interface

As the result of this work, training set was created, set of the class features was composed, classifier was chosen, trained and tested. The final and most efficient classifier is SGD classifier. The final tagger performance according to weighted F-score is 0.892 [10]. The client and the server are implemented for the client use.

Conclusions

The system that performs automatic markup of Ukrainian texts by parts of speech was designed, implemented and tested.

The main classification mechanism is based on supervised machine learning methods implemented in the de facto standard scikit-learn package. Effective use of the functions of this package involves the implementation of several research procedures.

In particular, for the implementation of the system, the search for the optimal set of class features and their construction from the training set was conducted. Also, the performance of various classification models was analyzed and the optimal classification model was selected, which turned out to be the SGD classifier. To obtain a high generalizing ability of the classifier, the K-block cross-validation technique was used.

The implemented POS tagger allows users to POS-tag large amounts of information in a short time and with high efficiency. Thus, the presented tagger can be used to create a corpus of the Ukrainian language, which will provide linguists with additional opportunities to study the Ukrainian language.

To improve the classification quality indicators, it seems promising to include contextual information in the feature space, consisting of information about tags of neighboring words.

References

1. Корпус української мови. URL: http://www.mova.info/syntaxis_search.aspx.
2. Проект створення корпусів текстів.Національний університет «Острозька академія» URL: https://www.oa.edu.ua/ua/departments/filologist/filol_literature/lexilab/project3.
3. Кочерган М.П. Вступ до мовознавства. К.: Академія, 2001. 368 с.
4. Manning C., Schütze H. Foundations of Statistical Natural Language Processing, Cambridge, MA: MIT Press, 1999. 717 p.
5. Zheng A., Casari A. Feature Engineering for Machine Learning. O’Reilly Media. 2018. 456 p.
6. Indurkhya N., Damerau F.J. Handbook of Natural Language Processing. Chapman and Hall. 2010. 702 p.
7. Scikit-learn: machine learning in Python – scikit-learn 0.23.2 documentation. URL: https://scikit-learn.org.
8. Refaeilzadeh P., Tang L., Liu H. Cross-Validation. Encyclopedia of Database Systems. Springer, 2009. P. 532-538.
9. Build cross-platform desktop apps with JavaScript, HTML, and CSS. URL: https://www.electronjs.org
10. Sokolova M., Japkowicz N., Szpakowicz S. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. *Advances in Artificial Intelligence, Lecture Notes in Computer Science*. 2006. V.4304. P. 1015-1021.

РОЗРОБКА ПРОГРАМНОЇ ПІДТРИМКИ ДЛЯ СТВОРЕННЯ КОРПУСУ УКРАЇНСЬКОЇ МОВИ І ЙОГО ВИКОРИСТАННЯ

О.І. Керпель, В.Г. Пенко

Одеський національний університет ім. І.І.Мечникова,
вул. Дворянська, 2, Одеса, 65000, Україна; e-mail: alexeykerpel15@gmail.com,
vpenko@onu.edu.ua

Актуальність роботи полягає в необхідності аналізу українських текстів з метою вивчення української мови і нестачі відповідних інформаційних і програмних ресурсів. Об'єкт дослідження - засоби складання POS-тегованих корпусів українських текстів. Предмет дослідження - процес розробки програмних засобів для створення корпусів українських текстів, зокрема, алгоритмів POS тегування. Мета роботи - дослідження і розробка програмних засобів для створення анотованого корпусу української мови. Для досягнення поставленої мети були вирішені наступні завдання: аналіз предметної області; вибір відповідних підходів і програмних засобів; створення навчальної множини; реалізація системи; навчання системи; тестування системи. Спроектовано та реалізовано підхід до генерації навчальної вибірки. Підхід заснований на використанні вже реалізованого тегера російської мови і схожості морфологічної будови слів російської та української мови. Для підвищення ефективності класифікації проведений пошук вдалої комбінації простору ознак і навчального алгоритму. Найбільш вдалими моделями машинного навчання для даного завдання виявилися стохастичний градієнтний спуск і дерева прийняття рішень. Для досягнення прийнятного рівня узагальнення використана перехресна перевірка. В результаті проведеної роботи було отримано навчений тегер української мови, який для вибірки літературних українських текстів забезпечує якість класифікації на рівні 0.892 по зваженій F-мірі. Реалізовано розподілений додаток з клієнт-серверною архітектурою, що дозволяє клієнтам здійснювати тегування власних текстів. При регулярному використанні цього додатку експертами-лінгвістами використану навчальну множину може бути покращено, що дозволить отримувати більш високі показники класифікації, що здійснює тегер.

Ключові слова: POS-тегування, корпус текстів, алгоритми машинного навчання, простір ознак.

РАЗРАБОТКА ПРОГРАММНОЙ ПОДДЕРЖКИ ДЛЯ СОЗДАНИЯ КОРПУСА УКРАИНСКОГО ЯЗЫКА И ЕГО ИСПОЛЬЗОВАНИЯ

А.И. Керпель, В.Г. Пенко

Одесский национальный университет им. И.И.Мечникова,
ул. Дворянская, 2, Одесса, 65000, Украина; e-mail: alexeykerpel15@gmail.com,
vpenko@onu.edu.ua

Актуальность работы заключается в необходимости анализа украинских текстов с целью изучения украинского языка и недостатке соответствующих информационных и программных ресурсов. Объект исследования – средства составления POS-тегированных корпусов украинских текстов. Предмет исследования – процесс создания программных средств для создания корпусов украинских текстов, в частности, алгоритмов POS тегирования. Цель работы – исследование и разработка программных средств для создания аннотированного корпуса украинского языка. Для достижения поставленной цели были решены следующие задачи: анализ предметной области; выбор подходящих подходов и программных средств; создание обучающего множества; реализация системы; обучение системы; тестирование системы. Спроектирован и реализован подход к генерации обучающей выборки. Подход основан на использовании уже реализованного теггера русского языка и сходстве морфологического строения слов русского и украинского языка. Для повышения эффективности классификации произведен поиск эффективной комбинации признакового пространства и обучающего алгоритма. Наиболее удачными моделями машинного обучения для данной задачи оказались стохастический градиент спуск и деревья принятия решений. Для достижения приемлемого уровня обобщения использована перекрестная проверка. В результате проведеной работы был получен обученный теггер украинского языка, который для выборки литературных украинских текстов обеспечивает качество классификации на уровне 0.892 по взвешенной F-мере. Реализовано распределенное приложение с клиент-серверной архитектурой, позволяющей клиентам осуществлять тегтрование собственных текстов. При регулярном использовании данного приложения экспертами-лингвистами использованное обучающее множество может быть улучшено, что позволит получать более высокие показатели классификации, осуществляемой теггером.

Ключевые слова: POS-тегирование, корпус текстов, алгоритмы машинного обучения, пространство признаков.