Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200619

Berri Corpus Manager: A Corpus Analysis Tool Using MongoDB Technology

Hugo SANJURJO-GONZÁLEZ [1]
Department of Information Technology, Electronics & Communications, University of Deusto, Spain

Abstract. Nowadays, there are many options for corpus linguistic analysis that make use of different approaches for corpus storage. There are tools based on SQL databases, dedicated implementations such as CQP/CWB and others that employ plain-text corpora. NoSQL databases have been widely used for big data, data mining and even sentiment analysis. However, as far as we can see, there is a lack of a widespread concordancer or consolidated framework that makes use of MongoDB architecture for the purposes of corpus linguistics. This paper aims to describe the architecture of a software that allows users to analyse monolingual and bilingual parallel corpora with grammatical annotation using MongoDB technology. Our premises are that MongoDB is ideal for non-structured data and provides high flexibility and scalability, so it may be also useful for corpus linguistic research. We analyse functionalities of MongoDB such as text search indexes and query format in order to examine its suitability.

Keywords. Corpus analysis tool, concordancer, NoSQL database, MongoDB, corpus linguistics

1. Introduction

There are many approaches for corpus storage regarding corpus analysis software. Sanjurjo-González [1] offers a comprehensive survey of the available corpus analysis software characterised by linguistic and technological features. In this survey, we can find tools based on SQL databases, mixed approaches that employ SQL, software that makes use of dedicated implementations for corpus indexing and querying [2] and last, software that employs plain-text corpora.

The most popular approaches for concordancers are those that are SQL-based, or dedicated implementations that use CQP/CWB [3]. For instance, SQL is used in corpus.bye.edu[2] and PELCRA[3], CQP/CWB is used in CQPWeb [4] and ACM[4] [5], among others. Although these approaches offer different performances using extreme scale cor-

[1] Corresponding Author: Hugo Sanjurjo-González; University of Deusto, Unibertsitate Etorbidea 24, 48007 Bilbao, Bizkaia, Spain; E-mail: hugo.sanjurjo@deusto.es

[2] http://corpus.byu.edu/

[3] http://nkjp.uni.lodz.pl/

[4] https://actres.unileon.es/wordpress/?page_id=663&lang=en

pora, there is no significant difference employing a smaller corpus for a linguistic oriented user carrying out the most common corpus linguistic operations such as KWIC (Key Word In Context) concordances, frequency lists, collocations, etc.

MongoDB[5] was officially released for production purposes in 2011 and is one of the most used NoSQL databases. It is a document-oriented database with high scalability and flexibility that stores data in BSON (binary JSON) documents. It is widely used to store data for sentiment analysis [6,7,8], data mining [9,10,11] or even big data [12,13,14]. In [15], Coole, Rayson and Mariani carried out different experiments using NoSQL databases, SQL databases and CWB/CQP with extreme scale corpus, showing that NoSQL databases like MongoDB or Cassandra are viable solutions for performing KWIC searches employing several servers. For this reason, their performance may be good enough using a large corpus even without clustering.

Consequently, this paper aims to describe the architecture of the software that allows users to analyse monolingual and bilingual corpora with grammatical annotation using MongoDB technology. Our premises are that MongoDB is ideal for non-structured information and corpus texts might be considered in this way, providing high flexibility and scalability.

The remainder of this article is organized as follows. In Section 2, we survey relevant research to the scope of the paper. We then describe the architecture and technology of the proposed software in Section 3. We introduce experiment results in Section 4. Finally, we describe conclusions in Section 5.

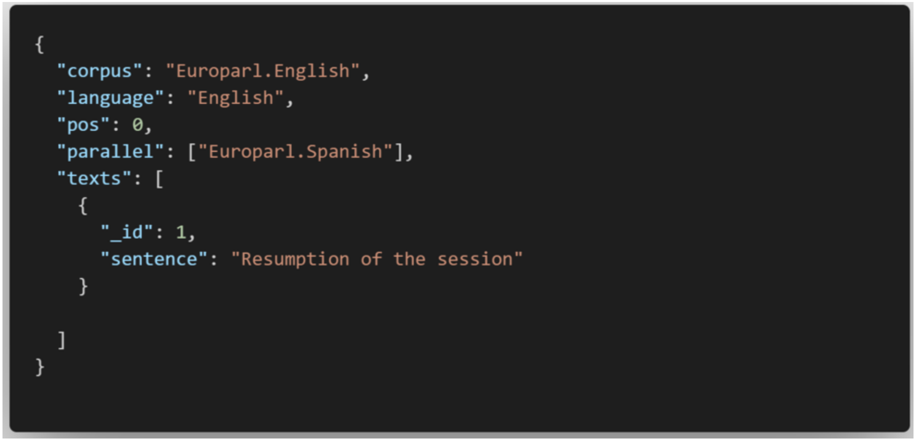
2. Related Work

NoSQL databases have gained popularity with emerging demands of scalable databases, mainly related to big data research [16]. This emerging trend of NoSQL paradigm for handling big data information systems should be at least considered for corpus linguistics research. Our premises are that MongoDB is ideal for non-structured information such as corpus texts thanks to its high flexibility, vertical scalability and schemmaless architecture. However, as far as we can see, there is no widespread concordancer or consolidated framework that makes use of MongoDB architecture for the purposes of corpus linguistics. In fact, some approaches make use of MongoDB for ad hoc solutions. We can mention Perkins [17] that presents a proof of concept that combines NLTK [18] with MongoDB database; Frey, Glaznieks and Stemle [19] download and merge data as a corpus zero using MongoDB; Gutierrez-Vasques, Sierra and Pomp [20] combine MongoDB with Lucene/Solr[6] to build and query their corpus; and last, Dorantes et al [21], develop a search interface using custom Python scripts and MongoDB database without describing any additional details about the implementation.

As previously mentioned, the most relevant work using MongoDB technologies in corpus analysis may be Coole, Rayson and Mariani [15]. In this case, they made their experiments in a clustering environment as a consequence of the extreme size of the corpus. In the present work, we do not use cluster components as the size of a typical corpus does not require as much power. In addition, we want to test MongoDB technology in the

[5] https://www.mongodb.com/

[6] https://lucene.apache.org/



```
{
  "corpus": "Europarl.English",
  "language": "English",
  "pos": 0,
  "parallel": ["Europarl.Spanish"],
  "texts": [
    {
      "_id": 1,
      "sentence": "Resumption of the session"
    }

  ]
}
```

Figure 1. Data model of Berri Corpus Manager

most common software development environment, composed of only one server, which does not require complex configurations or high hardware specifications.

3. Architecture and Technology

3.1. Data Model

Berri Corpus Manager employs MongoDB for corpus storage. For each corpus, we store language, size, wheter it includes any type of annotations or if it is part of a parallel corpus (Figure 1). As a consequence of the flexibility and schema-less architecture of the database, the design is very simple and can be easily modified if required. Grammatical annotations are included using prefix notation. Last, we employ sentence alignment for parallel corpora.

3.2. User Interface

User interface provides a way to query the corpus. This interface is based on Sanjurjo-González and Izquierdo [22] and offers a user-friendly layout that enables effective corpus analysis and simplifies the pattern (Figure 2) and parallel queries (Figure 3) that may or may not include annotations.

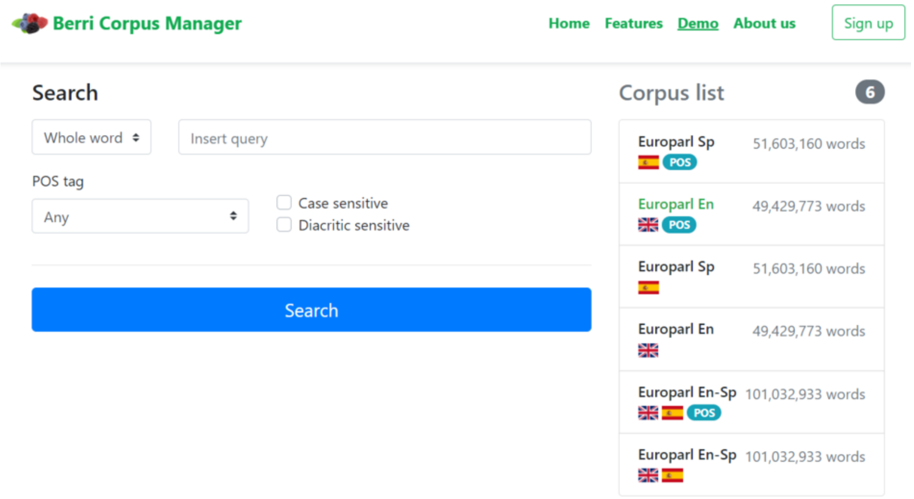
3.3. Implementation

To implement the software, we employ Python technologies, more concretely Flask[7] and PyMongo.[8]

Flask is a lightweight micro-framework based on Werkzeug web server and Jinja2 templates. It allows us to employ Python language to develop web applications. Flask maps url to different Python functions. Thus we can use Python in the server for natural

[7] https://flask.palletsprojects.com/en/1.1.x/

[8] https://docs.mongodb.com/drivers/pymongo



Whole word

Insert query

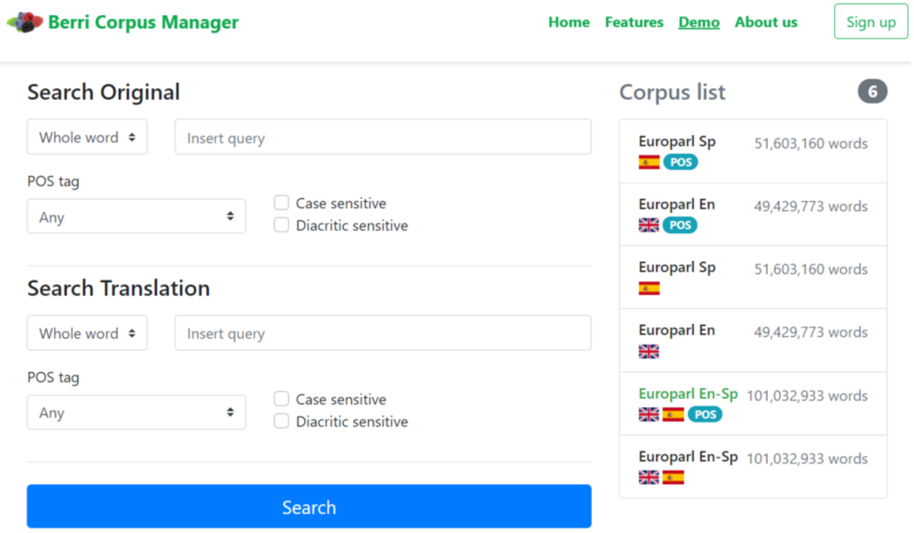
POS tag: Any

□ Case sensitive
□ Diacritic sensitive

Corpus list 6

Europarl Sp POS	51,603,160 words
Europarl En POS	49,429,773 words
Europarl Sp	51,603,160 words
Europarl En	49,429,773 words
Europarl En-Sp POS	101,032,933 words
Europarl En-Sp	101,032,933 words

Figure 2. User interface for monolingual queries



Search Original

Whole word

Insert query

POS tag: Any

□ Case sensitive
□ Diacritic sensitive

Search Translation

Whole word

Insert query

POS tag: Any

□ Case sensitive
□ Diacritic sensitive

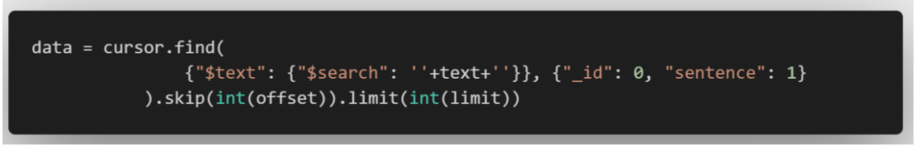
Corpus list 6

Europarl Sp POS	51,603,160 words
Europarl En POS	49,429,773 words
Europarl Sp	51,603,160 words
Europarl En	49,429,773 words
Europarl En-Sp POS	101,032,933 words
Europarl En-Sp	101,032,933 words

Figure 3. User interface for bilingual parallel queries

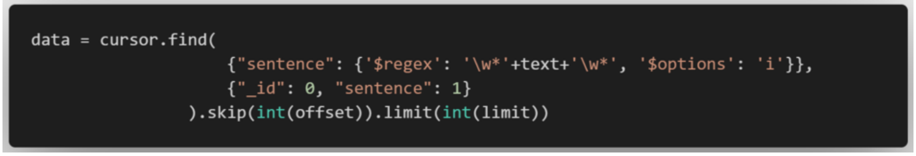
language processing tasks such as grammatical tagging, tokenisation and alignment, as well as for scripting tasks such as formatting and document corpus selection.

PyMongo is a Python distribution containing tools to work with MongoDB. By PyMongo we can easily interact with the MongoDB database using Python. As a consequence, we populate and query corpora employing available methods. It is also necessary to parse queries from the user interface to the MongoDB query language. To do that, we make use of the text search feature of MongoDB query language that includes a text index. In Figure 4, text variable refers to the query including or not including annotations.



```
data = cursor.find(
            {"$text": {"$search": ''+text+''}}, {"_id": 0, "sentence": 1}
        ).skip(int(offset)).limit(int(limit))
```

Figure 4. PyMongo code for full text search



```
data = cursor.find(
                {"sentence": {'$regex': '\w*'+text+'\w*', '$options': 'i'}},
                {"_id": 0, "sentence": 1}
            ).skip(int(offset)).limit(int(limit))
```

Figure 5. PyMongo code for regex search

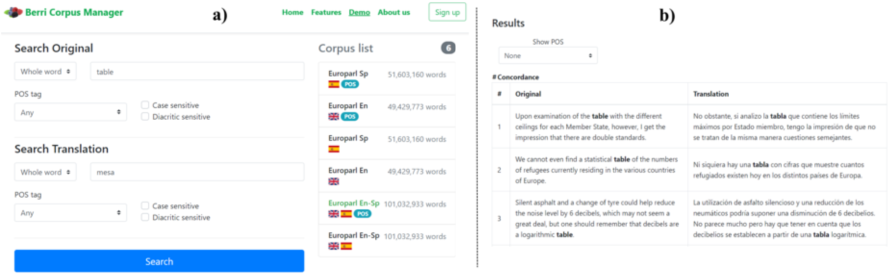
As it would be expected, text search increases performance, but it also has several issues regarding the most common queries in corpus linguistics:

1. It provides language specification, however, it uses stemming and removes stop words, so it might be useless if the user queries for a particular phrase of word. For this reason, it is better not to employ language specification. Size of the database increases significantly for this selection.
2. Text indexes do not support partial word searches, so they cannot be used to search patterns such as prefixes, suffixes, or others. To overcome this issue, we must employ $regex operator, which affects negatively the performance of the query (Figure 5).
3. As a consequence of the designed data model, grammatical annotations are included using prefix notation and some queries may be affected, for instance, if a user wants to search for all the words that are proper or common names.
4. As is the case with all the databases that are not linguistically oriented, the count function returns a count of documents, in our case sentences, that would match a query. However, one sentence may include more than one instance, so we need an additional processing. To do that, we make use of the MongoDB aggregation pipeline functionality. This functionality allows us to create a framework for data aggregation modelled on the concept of data processing pipelines. It works relatively well but it takes one minute or more if the number of results is high.

Parallel queries functionality of Berri Corpus Manager supports queries in both subcorpora at the same time in order to search the correspondences. For instance, we may be interested in searching what type of expressions translate the English word ”table” into Spanish (`tabla` or `mesa`). As MongoDB is not designed for this type of parallel queries, we search in both subcorpora at the same time, and obtain sentences for which identifiers are common in both results (Figure 6).

3.4. System Architecture

Berri Corpus manager operates across a simple Model-View-Controller pattern (Figure 7): Flask (controller), MongoDB (model), Jinja2 and Bootstrap v.4 (view). Berri Corpus Manager interacts with the corpus by means of PyMongo utilities. Python scripts are



Search Original

Whole word ⬍ table

POS tag: Any ⬍

□ Case sensitive
□ Diacritic sensitive

Search Translation

Whole word ⬍ mesa

POS tag: Any ⬍

□ Case sensitive
□ Diacritic sensitive

Search

Corpus list 6

Europarl Sp 🇪🇸 POS	51,603,160 words
Europarl En 🇬🇧 POS	49,429,773 words
Europarl Sp 🇪🇸	51,603,160 words
Europarl En 🇬🇧	49,429,773 words
Europarl En-Sp 🇬🇧 🇪🇸 POS	101,032,933 words
Europarl En-Sp 🇬🇧 🇪🇸	101,032,933 words

Results

Show POS: None ⬍

Concordance

#	Original	Translation
1	Upon examination of the **table** with the different ceilings for each Member State, however, I get the impression that there are double standards.	No obstante, si analizo la **tabla** que contiene los límites máximos por Estado miembro, tengo la impresión de que no se tratan de la misma manera cuestiones semejantes.
2	We cannot even find a statistical **table** of the numbers of refugees currently residing in the various countries of Europe.	Ni siquiera hay una **tabla** con cifras que muestre cuantos refugiados existen hoy en los distintos países de Europa.
3	Silent asphalt and a change of tyre could help reduce the noise level by 6 decibels, which may not seem a great deal, but one should remember that decibels are a logarithmic **table**.	La utilización de asfalto silencioso y una reducción de los neumáticos podría suponer una disminución de 6 decibelios. No parece mucho pero hay que tener en cuenta que los decibelios se establecen a partir de una **tabla** logarítmica.

Figure 6. Results of a parallel concordance

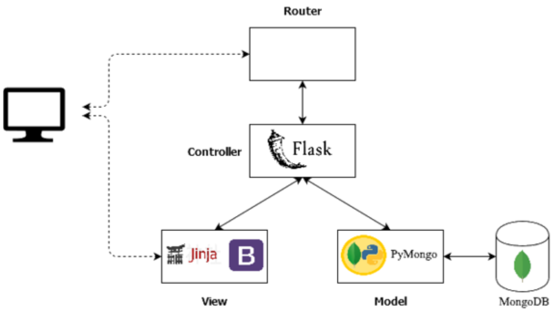


Figure 7. Architecture of Berri Corpus Manager

used to parse user queries into understandable commands using MongoDB format as well as natural language processing tasks.

4. Experiments

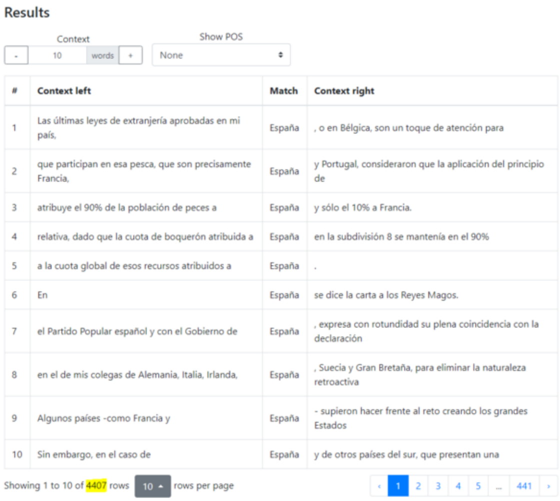
Berri Corpus Manager is able to handle monolingual and bilingual corpora with grammatical annotation. For our experiment, we selected the well-known corpus Europarl [23]. More concretely, we employed English and Spanish subcorpora that were already aligned and have a size of 49,429,773 and 51,603,160 words, respectively. Grammatical annotation has been carried out using Spacy. [9]

MongoDB presents a good performance if the search makes use of the text search index. It takes more time if it makes use of regex patterns. For instance, in our experiments, searching for all the occurrences of the word ”a” takes 20 seconds using the text index and 30 seconds if it is not used.

Berri Corpus Manager employs server pagination, so there are no issues returning results. However, as it was previously mentioned, returning the total number of matches might take some time if there is a high number of results.

Last, it should be mentioned that query plan cache methods save a lot of time for recurrent operations. Therefore, high performance queries can takes less time than expected if they have been previously executed.

[9] https://spacy.io/



Results

#	Context left	Match	Context right
1	Las últimas leyes de extranjería aprobadas en mi país,	España	, o en Bélgica, son un toque de atención para
2	que participan en esa pesca, que son precisamente Francia,	España	y Portugal, consideraron que la aplicación del principio de
3	atribuye el 90% de la población de peces a	España	y sólo el 10% a Francia.
4	relativa, dado que la cuota de boquerón atribuida a	España	en la subdivisión 8 se mantenía en el 90%
5	a la cuota global de esos recursos atribuidos a	España	.
6	En	España	se dice la carta a los Reyes Magos.
7	el Partido Popular español y con el Gobierno de	España	, expresa con rotundidad su plena coincidencia con la declaración
8	en el de mis colegas de Alemania, Italia, Irlanda,	España	, Suecia y Gran Bretaña, para eliminar la naturaleza retroactiva
9	Algunos países -como Francia y	España	- supieron hacer frente al reto creando los grandes Estados
10	Sin embargo, en el caso de	España	y de otros países del sur, que presentan una

Showing 1 to 10 of 4407 rows

Figure 8. Results of a monolingual concordance

5. Conclusions

This paper presented one of the first implementations of a concordancer using MongoDB technology. Employing lightweight technologies such as Flask makes development extremely quick. MongoDB has several advantages: it is schema-less, flexible and scalable. However, as it has not been designed for specific purposes of corpus linguistics, it also presents some issues, for instance, the unavailability of the partial search using text indexes, which has a serious effect on the queries' performance, or in returning the number of occurrences of a concordance, as it is based on the number of documents that match the query instead of the number of occurrences of the query. If these issues are solved, MongoDB can become a prominent technology for corpus linguistic software development as a consequence of its flexibility and fast deployment.

References

[1] Sanjurjo-González H. Desarrollo de un framework para el tratamiento de corpus lingüísticos [Development of a framework for corpus linguistic analysis] [Doctoral dissertation on the Internet). León, Spain: University of León; [cited 2020 Apr. 22]. Available from: http://hdl.handle.net/10612/6920

[2] McEnery T, Hardie A. Corpus Linguistics: Method, Theory and Practice. Cambridge University Press; 2012. 312p.

[3] Christ O. A modular and flexible architecture for an integrated corpus query system. In: Kiefer F, Kiss G, Pajzs J. editors. Proceedings of the 3rd International Conference on Computational Lexicography; 1994 Jul 7-10; Budapest, Hungary. Research Institute for Linguistics, Hungarian Academy of Sciences; p. 23-32.

[4] Hardie A. CQPweb—combining power, flexibility and usability in a corpus analysis tool. International Journal of Corpus Linguistics. 2012; 17(3):380-409.

[5] Sanjurjo-González H. Desarrollo de un framework para el tratamiento de corpus lingüísticos [Development of a framework for corpus linguistic analysis]. University of León; 2018. 116 p.

[6] Bai A, Hammer H, Yazidi A, Engelstad P. Constructing Sentiment Lexicons in Norwegian from a Large Text Corpus. In: Proceedings of the IEEE 17th International Conference on Computational Science and Engineering; 2014 Dec 19-21; Chengdu, China. IEEE Computer Society; p. 231-37.

[7] Gkontzis AF, Karachristos CV, Panagiotakopoulos CT, Stavropoulos EC, Verykios VS. Sentiment analysis to track emotion and polarity in student fora. In: Proceedings of the 21st Pan-Hellenic Conference on Informatics ; 2017 Sept 28-30; Larissa, Greece. The Association for Computing Machinery; p. 1-6.

[8] Ramzan M, Mehta S, Annapoorna E. Are tweets the real estimators of election results? In: Proceedings of the Tenth International Conference on Contemporary Computing (IC3); 2017 Aug 10-12; Noida, India. IEEE. p. 1-4

[9] Niekler A, Wiedemann G, Heyer G. Leipzig Corpus Miner–A Text Mining Infrastructure for Qualitative Data Analysis. In: Proceedings of the 11th Terminology and Knowledge Engineering 2014 proceedings, 39-47 2014 Jun 19-21; Berlin, Germany. Asociación Española de Terminología. p. 39-47.

[10] Santhiya K, Bhuvaneswari V. An automated MapReduce framework for crime classification of news articles using MongoDB. International Journal of Applied Engineering Research. 2018; 13(1):131-36.

[11] Staar PW, Dolfi M, Auer C, Bekas C. Corpus Conversion Service: A machine learning platform to ingest documents at scale. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2018 19-23 Aug; London, United Kingdom. The Association for Computing Machinery; p. 774-82.

[12] Rakib TBA, Soon LK. Using the Reddit Corpus for Cyberbully Detection. In: Nguyen N, Hoang D, Hong TP, Pham H, Trawiński B. editors. Proceedings of the 10th Asian Conference Intelligent Information and Database Systems (ACIIDS 2018); 2018 Mar 19-21; Dong Hoi City, Vietnam. Cham (Switzerland): Lecture Notes in Computer Science; 10751. p. 180-189.

[13] Kang Y. Park, I, Rhee J, Lee Y. MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data. IEEE Sensors Journal. 2016; 16(2):485-97.

[14] Plugge E. Hows D, Membrey P, Hawkins T. The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB. Apress; 2015. 336 p.

[15] Coole M, Rayson P, Mariani J. Scaling out for extreme scale corpus data. In: Proceedings of the 2015 IEEE International Conference on Big Data (Big Data). 2015 29 Oct - Nov 1; Santa Clara, CA. IEEE Computer Society. p. 1643-1649.

[16] Lee M, Jeon S, Song M. Understanding user's interests in nosql databases in stack overflow. In: Proceedings of the 7th International Conference on Emerging Databases. 2017 Aug 7-9; Busan, South Korea. Berlin (Germany): Springer. p. 128-137

[17] Perkins J. Python 3 text processing with NLTK 3 cookbook. Packt Publishing Ltd; 2014. 304 p.

[18] Bird S, Klein E. Loper E. Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit. O'Reilly Media Inc; 2009. 504 p.

[19] Frey, JC, Glaznieks A, Stemle EW. The DiDi Corpus of South Tyrolean CMC Data. In: Proceedings of the 2nd Workshop on Natural Language Processing for Computer-Mediated Communication/Social Media at GSCL2015 (NLP4CMC2015); 2005 Sep 25; Essen, Germany: German Society for Computational Linguistics & Language Technology; p. 1-6.

[20] Gutierrez-Vasques X, Sierra G, Pompa IH. Axolotl: a web accessible parallel corpus for spanish-nahuatl. In: Calzolari, N, Choukri K, Declerck T, Goggi S, Grobelnik M, Maegaard B, Mariani J, Mazo H, Moreno A, Odijk J, Piperidis S, editors. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) 2016 May 23-28: Portorož, Slovenia: European Language Resources Association; p. 4210-4.

[21] Dorantes A, Sierra G, Pérez TYD, Bel-Enguix G, Rosales MJ. Sociolinguistic corpus of whatsapp chats in spanish among college students. In: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media. 2018 July 20; Melbourne, Australia: Association for Computational Linguistics; p. 1-6.

[22] Sanjurjo-González H, Izquierdo M. P-ACTRES 2.0: A parallel corpus for cross-linguistic research. In: Doval I, Sánchez-Nieto MT, editors. Parallel Corpora for Contrastive and Translation Studies: New resources and applications 90. Amsterdam/Philadelphia: John Benjamins; 2019. p. 215-31.

[23] Koehn P. Europarl: A parallel corpus for statistical machine translation. In: Proceedings of Machine Translation Summit X Vol. 5. 2005 Sep 12-15; Phuket, Thailand. Asia-Pacific Association for Machine Translation. p. 79-86.