INTELCOMP PROJECT

A COMPETITIVE INTELLIGENCE CLOUD/HPC PLATFORM FOR AI-BASED STI POLICY MAKING

(GRANT AGREEMENT NUMBER 101004870)

D3.1. Scalable NLP pipelines

Deliverable information	
Deliverable number and name	D3.1. Scalable NLP pipelines
Due date	Dec 31, 2021
Delivery date	Dec 24, 2021
Work Package	WP3
Lead Partner for deliverable	Barcelona Supercomputing Center (BSC)
Authors	Aitor Gonzalez Aguirre, BSC Joan Llop Palao, BSC Marc Pàmies Massip, BSC Marta Villegas, BSC
Reviewers	Jesús Cid Sueiro, UC3M Haris Papageorgiou, ARC
Approved by	Jerónimo Arenas García, UC3M
Dissemination level	Confidential
Version	1.1

Table 1. Document revision history

Issue Date	Version	Comments
Nov 23, 2021	0.1	Initial draft
Dec 23, 2021	0.2	Submitted draft
Jan 17, 2022	1.0	Revised version
Jan 24, 2022	1.1	Final version
Jan 28, 2022	1.2	Changes requested by TM

DISCLAIMER

This document contains a description of the **IntelComp** project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium coordinator for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

The content of this publication is the sole responsibility of **IntelComp** consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 27 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors.

(http://europa.eu.int/)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101004870.

Contents

Figures

Tables

Acronyms

AI	Artificial Intelligence
ARC	Athena Research Center
BSC	Barcelona Supercomputing Center
CNN	Convolutional Neural Network
HDFS	Hadoop Distributed File System
HPC	High Performance Computing
GPU	Graphics Processing Unit
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
NMT	Neural Machine Translation
POS	Part-of-speech
UC3M	University Carlos III of Madrid
STI	Science, Technology and Innovation
WP	Work Package

Executive Summary

The Intelcomp NLP pipeline can be defined as a collection of tools that apply the requested transformations to unstructured textual data, which will be used by the Intelcomp services (document classification, subcorpus generation, topic modeling, etc.) as a preliminary step to process the datasets of interest. It has been designed to carry out standard text preprocessing tasks (e.g. *n*-grams detection, keywords extraction, lemmatization, etc) in a High Performance Computing environment, allowing the efficient and scalable processing of large amounts of documents. The final version of the pipeline will be deployed over the HPC infrastructure provided by the Barcelona Supercomputing Center and fully integrated with Intelcomp's Data Space. This document should serve not only as a report of the work performed by the members of WP3 but also as a complete guide for the targeted users and operators of the pipeline.

1. Introduction

The aim of this document is to present the Natural Language Processing (NLP) pipeline that will be used to preprocess the core datasets of the Intelcomp project. The pipeline includes a series of initial processing steps that will be applied to the primary datasets hosted by the Science, Technology and Innovation (STI) Data Space so that it can be fed to several services, including document classification, topic modeling and time analysis, among others. So it could be considered one of the basic pillars of the project, as its proper functioning is of vital importance to ensure a high-quality service from the final product.

In the sections that follow, the main components of the pipeline will be explained in detail, as well as the instructions on how to use them. In addition, this deliverable acts as a reference guide on the NLP pipeline usage. The software is also provided as a downloadable dockerized application facilitating reviewing and further experimentation. The docker file can be obtained from this publicly available repository[1].

2. Software

The underlying code has been written using the Python programming language. The main library used is Spark-NLP (Kocaman and Talby, 2021), an open-source NLP library that is natively built on Apache Spark and TensorFlow and can be seen as an extension of Spark's Machine Learning package. The library is known to be widely used among enterprises, mainly due to the fact that it provides simple, performant and accurate NLP annotations for Machine Learning pipelines. But most importantly, what makes it different from similar general-purpose libraries (e.g., spaCy, NLTK or CoreNLP) is its ease of scaling up in a distributed setup when dealing with huge amounts of data. This characteristic makes it especially suitable for the Intelcomp project, and the main reason why it was chosen after considering several alternatives. Thanks to Spark-NLP's optimizations, the pipeline can run orders of magnitude faster while still providing state-of-the-art results.

3. Supported languages

The pipeline supports a total of three languages: English, Spanish, and Greek. Every language-specific training has been carried out in a completely offline mode during the development phase, and the training scripts are available, so they can be used in the future with new data.

There are different processing options catering for diverse working scenarios in IntelComp. Depending on the operation requirements, datasets written in non-English can be automatically translated to English following the execution of the English pipeline, or the appropriate language pack is activated executing the language-dependent processing steps.

[1] https://hub.docker.com/r/bsctemu/wp3_nlp_pipelines.

4. USAGE

This section provides instructions on how to use the pipeline, including descriptions of the input/output format and the settings that can be configured by the user.

4.1. Docker image

The NLP pipeline has been developed and tested in BSC's clusters. The final platform will have a configurable amount of HPC nodes with a secure mechanism to be accessed from the outside. At the current stage, clusters cannot be accessed from the outside.

To facilitate the testing and deployment of the NLP pipelines, we have created a dockerized version that provides the ability to run the pipeline in a loosely isolated environment. The docker can be installed with the following command:

```
docker pull bsctemu/wp3_nlp_pipelines
```

4.2. Input data

The IntelComp Data Space facilitates the identification and retrieval of specific fields of unstructured text from several datasets. Most of these datasets are stored as distributed dataframes. Spark natively supports read/write operations to files in distriburted file systems such as Hadoop Distributed File System (HDFS), therefore the NLP pipeline should be easily integrated with the data lake. Data sets stored in other formats such as plain text documents or relational databases can also be easily distributed in a Spark cluster. A temporary input/output mechanism is provided enabling testing of the NLP pipeline with custom data files: the so-called docker volumes (described below). However, IntelComp integration tasks starting in M13 will produce a fully automated workflow, enabling IntelComp operators to activate the NLP pipeline on parts of specific datasets provided by the Data Provisioning subsystem, which is part of the IntelComp Data Space.

The docker volumes allow the user to mount and link a folder from the local system to the docker container of the pipeline. All files in the volume (directory in the local system) will be accessible inside the docker container. Users can instantiate the NLP pipeline with a custom volume as follows:

```
docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines
```

For testing purposes, input data can be loaded as a list of specific files with the option `--data_files`. Once loaded, the data is represented with a spark dataframe, where each row corresponds to an input file.

Custom Spark-NLP models can be loaded by modifying the config.json file, which must have the following format:

```
{
        "lang_pipe":     "path_to_lang_pipe",
        "jars":  {
                "cpu":   "path_to_cpu_spark_jar",
                "gpu":   "path_to_gpu_spark_jar"
        },
        "langs": {
                "lang_code":     {
                        "lang_name":              "language",
                        "embed_model_type":       "model_type",
                        "embeddings":             "path_to_embeddings_model",
                        "ner_model_type":         "model_type",
                        "ner":                    "path_to_ner_model",
                        "lemmatizer":             "path_to_lemmatizer_model",
                        "stopwords":              "path_to_stopwords"
                }
        }
}
```

An example of command to load all .txt files to a spark dataframe would be:

```
docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt
```

4.3. Output data

Each operation applied to the spark dataframe will result in one extra column for each input file. Once all required operations have been executed, the Spark dataframe will contain all the results and can be uploaded back to the IntelComp Data Space or used as the input to other WP3 AI services. For testing purposes, the standalone version that we provide at this stage stores the final dataframe into a CSV in the specified path.

Note that the CSV path must be inside the volume. An example of usage to save the final dataframe as a CSV would be:

```
>>> docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt --output_csv
/home/wp3_nlp_pipelines/data/volume/spark_df_out.csv

Number of documents loaded into dataframe: 3
+--------------------+--------------------+
|                text|            filepath|
+--------------------+--------------------+
|Conozca el papel ...|file:///home/wp3_...|
|Μάθετε για τον ρό ...  |file:///home/wp3_...|
|Learn about the E...|file:///home/wp3_...|
+--------------------+--------------------+
```

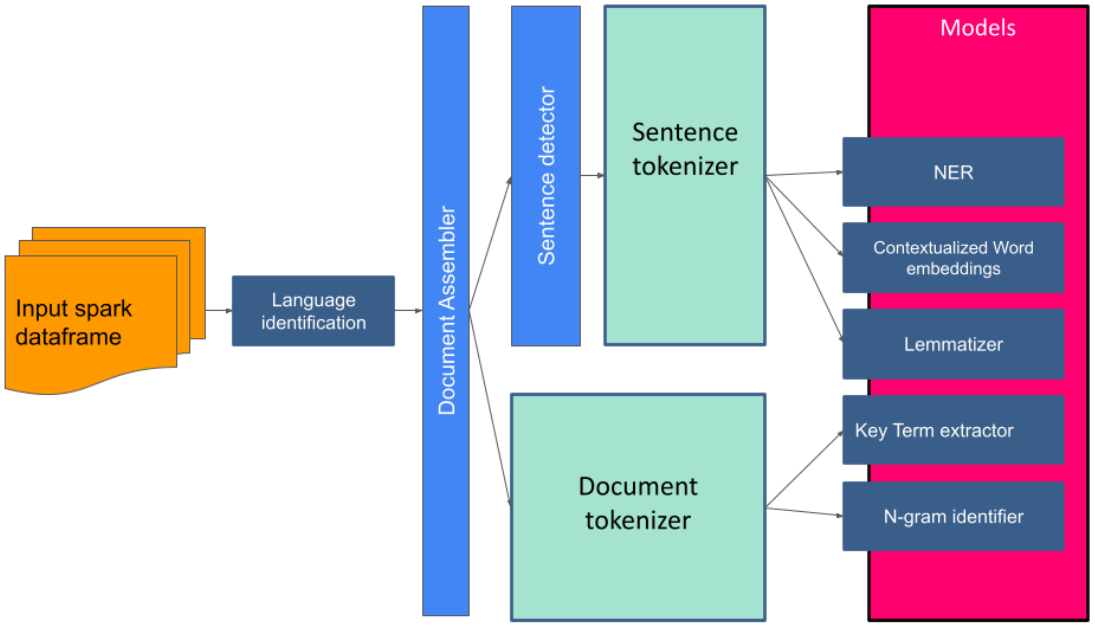
```
+--------------------+--------------------+--------+
|                text|            filepath|language|
+--------------------+--------------------+--------+
|Conozca el papel ...|file:///home/wp3_...|    [es]|
|Μάθετε για τον ρό ...   |file:///home/wp3_...|    [el]|
|Learn about the E...|file:///home/wp3_...|    [en]|
+--------------------+--------------------+--------+
```

The output will be stored in a CSV file of name `spark_df_out` in `/path/to/data_dir/`

5. Pipeline components

The NLP pipeline depicted in Figure 5-1 and presented in the following subsections is composed of a series of data processing components that are executed sequentially. Its modular design allows it to add new components without affecting the rest. In the same way, it is possible for the user to deactivate some of the components through command line arguments, so that the total runtime can be significantly reduced by skipping unnecessary steps.

Figure 5-1. Data processing blocks



The main components of the pipeline are described below.

5.1. Language identification

The first functionality of the pipeline is to automatically detect the language of each document based on its content. Already pretrained models from Spark-NLP are used for this task, which involve CNN architectures trained on large datasets, such as Wikipedia and Tatoeba. The language identification component behaves as a single-class classifier, meaning that the output for a given text will be one and only one of the possible languages. If the input text is written in a language that is not supported by the pipeline, the corresponding document will be ignored, assuming it as a negligible error since it will be the case of the great minority of documents. This means that the output dataframe will not contain a row for that specific document.

The outcome will be the input dataframe with an additional column called “language”. Each cell of this new column will contain a string that can take one of three possible values: “en”, “es” and “el”, representing the three languages supported by the pipeline.

```
>>> docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt --output_csv
/home/wp3_nlp_pipelines/data/volume/spark_df_out.csv

Number of documents loaded into dataframe: 3
+--------------------+--------------------+
|                text|            filepath|
+--------------------+--------------------+
|Conozca el papel ...|file:///home/wp3_...|
|Μάθετε για τον ρό ...  |file:///home/wp3_...|
|Learn about the E...|file:///home/wp3_...|
+--------------------+--------------------+

+--------------------+--------------------+--------+
|                text|            filepath|language|
+--------------------+--------------------+--------+
|Conozca el papel ...|file:///home/wp3_...|    [es]|
|Μάθετε για τον ρό ...  |file:///home/wp3_...|    [el]|
|Learn about the E...|file:///home/wp3_...|    [en]|
+--------------------+--------------------+--------+
```

The output will be stored in a CSV file of name `spark_df_out` in `/path/to/data_dir/`

5.2. *n*-grams detection

If activated, this component of the pipeline converts the content of the input documents into an array of *n*-grams, where the value of *n* is specified by the user. For instance, if the value of *n* is set to 3, then the pipeline returns all possible unigrams, bigrams and trigrams. If set to 4, it will also include four-grams, and so on so forth. Note that this is not related to the keyword extractor (see next section), so, rather than returning meaningful *n*-grams, it is limited to the

generation of all possible combinations of consecutive words, always in accordance with the parameters provided by the user to control the length of the output sequences.

By default, each *n*-gram is represented by an underscore-separated string of words, unless the user specifies another separator by a command line argument.

The outcome will be the input dataframe with an additional column called “ngrams”. Each cell of this new column will contain a list of *n*-grams for the corresponding document.

```
>>> docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt --output_csv
/home/wp3_nlp_pipelines/data/volume/spark_df_out.csv --ngrams --grams_size 1

+--------------------+--------------------+--------+--------------------+
|                text|            filepath|language|              ngrams|
+--------------------+--------------------+--------+--------------------+
|Conozca el papel ...|file:///home/wp3_...|    [es]||[Conozca, el, pap...|
|Learn about the E...|file:///home/wp3_...|    [en]||[Learn, about, th...|
|Μάθετε για τον ρό ...  |file:///home/wp3_...|    [el]||[Μάθετε, για, τον ...  |
+--------------------+--------------------+--------+--------------------+
```

5.3. Keywords extraction

If activated, this component of the pipeline will extract the most important terms of each document. The unsupervised algorithm running under-the-hood is known as Yake (Campos et al., 2020), a feature-based system that is independent of the corpus, domain and language. In order to identify keywords from unstructured documents, it basically relies on local text features as well as statistical information (e.g., term co-occurrence and frequencies). Note that the outcome is based on the input document alone, so the unavailability of a training corpus is not a limitation.

The user can set the minimum and maximum number of tokens that a keyword can have, as well as the maximum number of keywords to be returned. Stopwords are filtered out by default. Note that this component is independent from the *n*-gram detector, mainly because the desired size of the grams does not necessarily have to be the same as the length of the keywords, but also because stopwords are kept intact in the *n*-gram detector’s output.

The outcome will be the input dataframe with an additional column called “keywords”. Each cell of this new column will contain a list of the top keywords sorted in accordance with the score assigned by the algorithm.

```
>>> docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt --output_csv
/home/wp3_nlp_pipelines/data/volume/spark_df_out.csv --keywords
--num_keywords 2
```

```
+--------------------+--------------------+--------+--------------------+
|                text|            filepath|language|            keywords|
+--------------------+--------------------+--------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[commissioners, p...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[comisarios, comi...|
|Μάθετε για τον ρό ...  |file:///home/wp3_...|    [el]| [της, επιτροπής, ...   |
+--------------------+--------------------+--------+--------------------+
```

5.4. Lemmatization

If activated, this component of the pipeline will transform each word from the input text to its canonical form. In other words, all variations of a lexeme will be mapped to the same canonical form, i.e., their lemma. Pretrained language-specific models from Spark-NLP's hub were used rather than loading specific dictionaries of predefined lemmata. If the default models are not appropriate, it is possible for the user to add custom models whenever necessary. To do so, the model path specified in the *config* file as "lemmatizer" has to be modified for each language to point to a valid Spark-NLP model.

The outcome will be the input dataframe with an additional column called "lemmas". Each cell of this new column will contain a list of lemmatized terms from the corresponding document.

```
>>> docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt --output_csv
/home/wp3_nlp_pipelines/data/volume/spark_df_out.csv --lemmatize

+--------------------+--------------------+--------+--------------------+
|                text|            filepath|language|              lemmas|
+--------------------+--------------------+--------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[learn, about, th...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[conocer, el, pap...|
|Μάθετε για τον ρό...  |file:///home/wp3_...|    [el]|[μάθετε, για, ο, ...  |
+--------------------+--------------------+--------+--------------------+
```

5.5. Word embeddings

If activated, this component of the pipeline will generate the word embeddings for every word in the input document. The numerical representation of each word is produced by transformer-based models (Vaswani, Shazeer et al., 2017) that have been pre-trained on a large corpus of language-specific data. By default, for English data we use the original RoBERTa model (Liu, et al., 2019), for Greek we use the so-called GreekBERT (Koutsikakis, et al, 2020) and for Spanish the RoBERTa-base version of the MarIA family of models (Gutiérrez-Fandiño, et al., 2021). If the default models are not appropriate, it is possible for the user to add custom models. To do so, the model path specified in the *config* file as "embeddings" has to be modified for each language to point to a valid Spark-NLP model.

The outcome will be the input dataframe with two additional columns: “tokens” and “embeddings”. Each cell of the former will contain a list of tokens, which basically are subword units that are recognized by the model’s tokenizer. And as for the latter, each cell of this new column will contain a list of lists, where each list contains the numerical representation of a single token. The length of these vectors of float numbers is 768 in all cases, and the length of the outer list depends on the number of tokens contained in the text.

```
>>> docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt --output_csv
/home/wp3_nlp_pipelines/data/volume/spark_df_out.csv --embeddings

+--------------------+--------------------+--------+--------------------+--------------------+
|                text|            filepath|language|              tokens|          embeddings|
+--------------------+--------------------+--------+--------------------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[Learn, about, th...|[[0.030536707, -0...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[Conozca, el, pap...|[[0.043688737, -0...|
|Μάθετε για τον ρό ...   |file:///home/wp3_...|    [el]|[Μάθετε, για, τον ...  |[[-0.85215056,
-0...|
+--------------------+--------------------+--------+--------------------+--------------------+
```

5.6. Named Entity Recognition (NER)

If activated, this component of the pipeline locates and classifies named entities in the original text. The default models identify entities that belong to any of the following predefined categories:

- Person (PER): Entity representing a human being regarded as an individual.
- Location (LOC): Entity representing a particular place or position.
- Organization (ORG): Entity representing one or more people with a particular purpose.
- MISCELLANEOUS (MISC): Denotes all entities that do not belong to another category.

The categories are model-dependent, so it is possible to detect other types of entities (e.g., health-related terms) by loading a model that has been trained on a specific domain.

This task uses transformer-based models as well. When the input contains English data, we use a RoBERTa-base model fine-tuned on a modified version of the NER dataset from CoNLL-2003 (Sang and De Meulder, 2003). For Greek we did not have any language-specific model at our disposal, so a multilingual one is used instead, a XLM-Roberta-base (Conneau and Lample, 2019) architecture that was fine-tuned on a total of 40 languages, achieving 0.89 F1-score when evaluated on 10,000 Greek documents from the XTREME benchmark (Hu et al., 2020). For Spanish we use a Spanish RoBERTa-base model (Gutiérrez-Fandiño, A. et al., 2021) fine-tuned on data from the CoNLL-2002 shared task (Sang, E. F., & De Meulder, F., 2002).

Custom models can be added by the user. To do so, the model path specified in the *config* file as “ner” has to be modified for each language to point to a valid Spark-NLP model.

The outcome will be the input dataframe with an additional column called “ner”. Each cell of this new column will contain a list of detected entities with their corresponding types.

```
>>> docker run -v /path/to/data_dir/:/home/wp3_nlp_pipelines/data/volume
bsctemu/wp3_nlp_pipelines python nlp_pipeline.py --data_files
/home/wp3_nlp_pipelines/data/volume/*.txt --output_csv
/home/wp3_nlp_pipelines/data/volume/spark_df_out.csv --extract_entities

+--------------------+--------------------+--------+--------------------+
|                text|            filepath|language|                 ner|
+--------------------+--------------------+--------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[European Commiss...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[Comisión Europea...|
|Μάθετε για τον ρό ...   |file:///home/wp3_...|   [el]| [Ευρωπαϊκής Επιτρ ...  |
+--------------------+--------------------+--------+--------------------+
```

6. Next steps

An efficient integration of all NLP pipelines with the IntelComp Data Space in coordination with WP5 is the next step in order to maximize the use of the BSC's resources. Furthermore, execution of the NLP pipelines is expected to be automatically instantiated over all core data sets in the Data Space, including new data sets that will be ingested during the project as a result of further WP2 activities and Living Labs needs, and updated versions of existing datasets.

Regarding languages, new additional languages could be integrated to the original pipeline. It is expected that the final version will support a total of 5 languages, namely: English, French, German, Greek and Spanish. In the coming months we will extend the list of languages, fine-tuning language-specific models for the tasks that require so.

If necessary, we might explore the possibility of tuning the pipelines for the Intelcomp selected domains, once the domain-specific subcorpus generation tasks are completed (IntelComp Task 3.3) and such subcorpus become available. If there is a requirement from other services, it would also be possible to include new functionalities such as part-of-speech tagging or chunking. Note that some intermediate steps (e.g. tokenization, stopword removal, etc) are performed internally even though they do not appear in the pipeline's outcome, but it would be easy to include them in the output if there is a need.

References

Kocaman, V., & Talby, D. (2021). Spark nlp: Natural language understanding at scale. *Software Impacts*, *8*, 100058.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, *509*, 257-289.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Koutsikakis, J., Chalkidis, I., Malakasiotis, P., & Androutsopoulos, I. (2020, September). Greek-bert: The greeks visiting sesame street. In *11th Hellenic Conference on Artificial Intelligence* (pp. 110-117).

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., ... & Villegas, M. (2021). Spanish language models. *arXiv preprint arXiv:2107.07253*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050.

Sang, E. F., & De Meulder, F. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In COLING-02: The 6th Conference on Natural Language Learning

Conneau, A., & Lample, G. (2019). Cross-lingual language model pretraining. Advances in Neural Information Processing Systems, 32, 7059-7069.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., & Johnson, M. (2020, November). Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning* (pp. 4411-4421). PMLR.

ANNEX I: SPARK ARGUMENTS

--log_level

Verbosity level of the log messages. Options: ALL, DEBUG, ERROR, FATAL, INFO, OFF, TRACE, WARN. The default value is OFF to avoid unnecessary information.

--use_gpu

If activated loads the GPU-computing spark version and will use all available gpus. It's only supported in the cluster deployment.

--driver_memory

Amount of memory to use for the driver process, 32G by default.

--max_result_size

Limit of total size of serialized results of all partitions for each Spark action in bytes. Should be at least 1M, or 0 for unlimited, 32G by default.

--master_url

The cluster manager to connect to. The default value runs Spark locally with as many worker threads as logical cores on your machine.

--buffer_max

Maximum allowable size of Kryo serialization buffer, in MiB unless otherwise specified. This must be larger than any object you attempt to serialize and must be less than 2048m. Increase this if you get a "buffer limit exceeded" exception inside Kryo.

--config_file

Path to Spark's configuration file and all components of the pipeline. The config file contains the paths to all resources. The dockerized version has the relative paths inside the docker and should not be modified unless a model is replaced.

--cache_folder

Location to extract pretrained Models and Pipelines.

--cluster_tmp_dir

Location to use on a cluster for temporary files such as unpacking indexes for WordEmbeddings. By default, this is the location of hadoop.tmp.dir set via Hadoop configuration for Apache Spark.

ANNEX II: PIPELINE ARGUMENTS

--data_files

List of data file paths to be processed.

--output_csv

Path of the output CSV file.

--lemmatize

Whether to lemmatize the content of each document.

--keywords

Whether to extract keywords from each document.

--min_ngrams_for_keywords

Minimum *n*-grams a keyword should have.

--max_ngrams_for_keywords

Maximum *n*-grams a keyword should have.

--num_keywords

Number of keywords to extract.

--extract_entities

Whether to perform named-entity recognition on each document.

--embeddings

Whether to extract word embeddings.

--ngrams

Whether to enable *n*-grams extraction.

--grams_size

the size of ngrams

--ngram_delimiter

Character to use to join tokens when N is greater than 1. Default '_'.

--verbose

Whether the program should be verbose or not. Default to True.

ANNEX III: EXECUTION EXAMPLE

```
docker run --rm -v /path/to/data/:/home/wp3_nlp_pipelines/data/volume bsctemu/wp3_nlp_pipelines python nlp_pipeline.py
--data_files /home/wp3_nlp_pipelines/data/eu_data_test --output_csv /home/wp3_nlp_pipelines/data/volume/spark_df_out.csv
--lemmatize --keywords --num_keywords 5 --extract_entities --embeddings --ngrams
```

```
Number of documents loaded into dataframe: 3
+--------------------+--------------------+
|                text|            filepath|
+--------------------+--------------------+
|Conozca el papel ...|file:///home/wp3_...|
|Μάθετε για τον ρό...  |file:///home/wp3_...|
|Learn about the E...|file:///home/wp3_...|
+--------------------+--------------------+

+--------------------+--------------------+--------+
|                text|            filepath|language|
+--------------------+--------------------+--------+
|Conozca el papel ...|file:///home/wp3_...|    [es]|
|Μάθετε για τον ρό...  |file:///home/wp3_...|    [el]|
|Learn about the E...|file:///home/wp3_...|    [en]|
+--------------------+--------------------+--------+

+--------------------+--------------------+--------+--------------------+
|                text|            filepath|language|              ngrams|
+--------------------+--------------------+--------+--------------------+
|Conozca el papel ...|file:///home/wp3_...|    [es]|[conocer, el, pap...|
|Μάθετε για τον ρό...  |file:///home/wp3_...|    [el]|[Μάθετε, για, τον... |
|Learn about the E...|file:///home/wp3_...|    [en]|[Learn, about, th...|
+--------------------+--------------------+--------+--------------------+

+--------------------+--------------------+--------+--------------------+--------------------+
|                text|            filepath|language|              ngrams|              lemmas|
+--------------------+--------------------+--------+--------------------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[Learn, about, th...|[Learn, about, th...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[Conozca, el, pap...|[Conozca, el, pap...|
|Μάθετε για τον ρό...  |file:///home/wp3_...|    [el]|[Μάθετε, για, τον...  |[Μάθετε, για, ο, ... |
+--------------------+--------------------+--------+--------------------+--------------------+

+--------------------+--------------------+--------+--------------------+--------------------+--------------------+
|                text|            filepath|language|              ngrams|              lemmas|            keywords|
+--------------------+--------------------+--------+--------------------+--------------------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[Learn, about, th...|[Learn, about, th...|[college, commiss...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[Conozca, el, pap...|[Conozca, el, pap...|[ue, políticas, c...|
|Μάθετε για τον ρό...  |file:///home/wp3_...|    [el]|[Μάθετε, για, τον... |[Μάθετε, για, ο, ...  |[της, επιτροπής, ...  |
+--------------------+--------------------+--------+--------------------+--------------------+--------------------+

+--------------------+--------------------+--------+--------------------+--------------------+--------------------+--------------------+
|                text|            filepath|language|              ngrams|              lemmas|            keywords|                 ner|
+--------------------+--------------------+--------+--------------------+--------------------+--------------------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[Learn, about, th...|[Learn, about, th...|[college, commiss...|[European Commiss...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[Conozca, el, pap...|[Conozca, el, pap...|[ue, políticas, c...|[Comisión Europea...|
|Μάθετε για τον ρό...  |file:///home/wp3_...|    [el]|[Μάθετε, για, τον...  |[Μάθετε, για, ο ... |[της, επιτροπής,...    |[Ευρωπαϊκής Επιτρ...  |
+--------------------+--------------------+--------+--------------------+--------------------+--------------------+--------------------+

+--------------------+--------------------+--------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|                text|            filepath|language|              ngrams|              lemmas|            keywords|                 ner|              tokens|          embeddings|
+--------------------+--------------------+--------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|Learn about the E...|file:///home/wp3_...|    [en]|[Learn, about, th...|[Learn, about, th...|[college, commiss...|[European Commiss...|[Learn, about, th...|[[0.030536707, -0...|
|Conozca el papel ...|file:///home/wp3_...|    [es]|[Conozca, el, pap...|[Conozca, el, pap...|[ue, políticas, c...|[Comisión Europea...|[Conozca, el, pap...|[[0.043688737, -0...|
|Μάθετε για τον ρό... |file:///home/wp3_... |    [el]|[Μάθετε, για, τον...  |[Μάθετε, για, ο,... |[της, επιτροπής,...    |[Ευρωπαϊκής Επιτρ...  |[Μάθετε, για, τον... |[[[-0.85215056, -0...|
+--------------------+--------------------+--------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
```