intelcomp

European Commission

INTELCOMP PROJECT

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

(GRANT AGREEMENT NUMBER 101004870)

D3.4. Classification Service

Deliverable information	
Deliverable number and name	D3.4. Classification Service
Due date	September 23, 2022
Delivery date	September 23, 2022
Work Package	WP3
Lead Partner for deliverable	Barcelona Supercomputing Center (BSC)
Authors	Aitor Gonzalez, BSC Joan Llop, BSC Marc Pàmies, BSC Marta Villegas, BSC Sotiris Kotitsas, ARC Dimitris Pappas, ARC
Reviewers	Doaa Samy, UC3M Sotiris Kotitsas, ARC
Approved by	Jeronimo Arenas, UC3M
Dissemination level	Public
Version	1.0

Table 1. Document revision history

Issue Date	Version	Comments
June 30, 2022	0.1	Initial draft for internal review
August 24, 2022	0.2	Revised Version
September 16, 2022	0.3	Revised version
September 23, 2022	1.0	Submitted version

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
BSC	Barcelona Supercomputing Center
CPU	Central Processing Unit
DoA	Description of Action
DOCDB	EPO worldwide bibliographic data
EPO	European Patent Office
EU	European Union
FOS	Field Of Science
HDFS	Hadoop Distributed File System
HPC	High Performance Computing
ICD	International Classification of Diseases
IPC	International Patent Classification
IPO	Intellectual Property Office
GPU	Graphics Processing Unit
LM	Language Model
MAG	Microsoft Academic Graph
ML	Machine Learning
NACE	Nomenclature of Economic Activities
NLP	Natural Language Processing
RTE	Recognizing Textual Entailment
UC3M	University Carlos III of Madrid
SDG	Sustainable Development Goals
STI	Science, Technology and Innovation
WIPO	World Intellectual Property Organization
WP	Work Package

Executive Summary

IntelComp is an EU-funded project whose main objective is the development of a cloud platform that facilitates evidence-based policy-making in the field of Science, Technology and Innovation (STI). Such platform will be used by public administrators from European countries to take intelligent-driven decisions backed by innovative AI services. The Machine Learning algorithms running in the back-end enable the extraction of useful insights from the data available in Intelcomp's Data Space, so that the end users can leverage this information to their advantage. The goal of this document is to provide an overall description of the taxonomical classification service (task 3.4 of the DoA) that will be used as a model trainer for new taxonomies, as well as information about a set of ready-to-use classifiers. The model trainer includes a series of functionalities so that new classifiers can be easily trained given any labeled dataset, as well as an evaluation workbench module that can be used to label new data. As an additional contribution, a zero-shot text classifier is provided to classify data into classes for which there is no training data available at the cost of lower performance. All the components of the classification service have been designed to run in a High Performance Computing environment, allowing efficient and scalable processing of large amounts of data. This document should serve not only as a report of the work performed by the members of WP3 but also as a user guide for the classification service, as it contains detailed explanations on how to train new classifiers on future datasets.

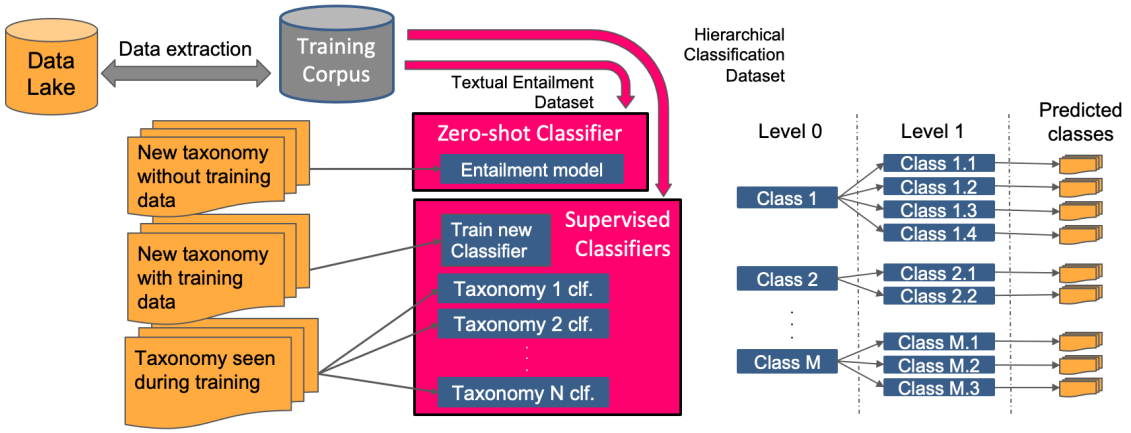
1. INTRODUCTION

The aim of this document is to present the Service for document taxonomical classification that will enable the extraction of useful insights from the data available in Intelcomp's Data Space. The service for document taxonomical classification within WP3 will be applied to the primary datasets hosted by the Data Space so that it can be fed to several services, including topic modelling and time analysis, among others.

This document describes in great detail the AI-based service of document taxonomical classification, which will be offered by the Intelcomp platform to guide data-driven policy making in the field of Science, Technology and Innovation (STI). The classification of documents into taxonomies is not only one of the core objectives of the Intelcomp project but also a fundamental step for understanding and analysing the substantial amount of dynamic and heterogeneous data (i.e. projects, reports, publications, patents, dissemination articles, etc.) produced by the STI field at all levels within the European Union.

This deliverable provides ready-to-use classifiers and, most importantly, a collection of training scripts that can be executed in HPC to create new classifiers whenever new data is ingested in the project's data space. The instructions to use these tools (available here[1]) are provided in this document, which should be used as a reference guide for future users of the model trainer.

Figure 1. Service for document taxonomical classification overview



As can be seen in Figure 1, the supervised classifiers have been trained on a variety of taxonomies available in the Data Lake. Taking advantage of the fact that most datasets have a hierarchical structure, a cascade of classifiers trained at each level of the hierarchy is used in order to maximize the final performance, otherwise, the high number of classes would significantly increase the complexity of the task. In addition, a zero-shot text classifier was trained to deal with classes for which no data is available, at the cost of lower performance than its supervised counterpart.

[1] https://github.com/IntelCompH2020/taxonomical-classification

The remainder of this document is structured as follows:

- Section 2 provides an overview of the main components of the classification service.
- Section 3 presents the datasets used to train different models.
- Section 4 defines the taxonomies of such datasets.
- Section 5 describes the different types of classifiers that have been trained.
- Section 6 concludes this deliverable with final remarks.

2. Service for document taxonomical classification

This section provides a general idea of how the classifiers work and explanations regarding the training process followed for their creation. Two deep learning classifiers have been trained on the PATSTAT and NACE2 taxonomies (more information in upcoming sections) and a multi-layer graph-based neural network has been trained on the Field Of Study (FOS) taxonomy.

2.1. Training

The deep learning classifiers were trained using classical supervised finetuning from a large language model. Any encoder model can be used as a starting point, with RoBERTa-large [4] being the architecture of choice for this project. The RoBERTa-large architecture was first proposed by Facebook AI and it is a transformer model trained on raw English text data without any labelled examples (self-supervised). More precisely, the pre-training of these models is done by randomly masking 15% of the words in the input, then running the sentence through the model and predicting the masked words. These differ from Recurrent Neural Networks (RNN) in the sense that an RNN looks at one word after the other, or from the autoregressive models that masked the future words. This approach allows the model to learn bidirectional representations of the words making these representations useful features to classify texts.

The input of each classifier is plain text and the output is a boolean vector with the length of the number of possible labels, a true value indicates that the text is classified as that label.

The problem at hand is known as multi-label muti-class text classification, which means that there are several classes and each document can belong to more than one class, for instance, one patent can be classified as computer science and applied mathematics at the same time. Training is done using labelled examples where each text has been associated with a boolean vector with the labels as previously described.

2.2. Dataloader

The first step to be able to either classify a document or train a new classifier is to load the data, regardless of the size and taxonomy used. The data is stored in parquet files and the data mediator that preprocesses and yields batches of samples is what we call the dataloader. It is composed of several components: *Init*, *Info* and *Generator of examples*. In order to have a unique dataloader for each possible input parquet, we need to build one that is robust enough to ingest different types of datasets. The key ideas implemented in each component of the dataloader are described below.

The *init* component

The dataloader reads any parquet table as train, dev or test. If only a test set is provided, then the set of unique labels will be empty. If dev or train are present and the label column is available, it iterates over the label column of all parquet files except the test ones and extracts the unique labels.

The *info* component

The info method returns the features that the model will use to train, evaluate and predict. In case training or dev splits are present, this method will return the feature "text" plus the labels which are represented as a vector with 1 in the positive labels and 0 in the other labels. In case only test is present, the only feature will be "text".

The *generator* component

This method reads each parquet for the train, dev or test splits and yields the examples with the essential features described in the info method. The key idea is to output each labelled example as a one-hot encoded vector in which we have a vector of size 1 x number of labels, and the labels assigned to that example are labelled as 1 and the rest are labelled as 0. In case only a test split is present, the generation of examples only yields the text column.

2.3. Evaluation

During the validation phase of training, the metrics are calculated by applying a sigmoid function to the predictions in order to have a value between zero and one for each label. All labels with a value higher or equal to 0.5 are considered positive. The F1 metric between the ground truth and the predicted labels is reported.

2.4. Software

The underlying code has been written using the Python programming language. The main library used is HuggingFace's Transformers [3], an open-source NLP library that consists of carefully engineered state-of-the-art Transformer architectures under a unified API. The library is known to be widely used among researchers and industry alike, mainly due to the fact that it provides easy-to-customize implementations of the state-of-the-art Transformers architectures.

Since the data space will be updated on a regular basis, the code to train a model on future taxonomies is provided in the taxonomical-classification GitHub repository, within the IntelComp organization repository[2]. Great effort has been put into simplifying the user's job, so that the finetuning process can be launched (either in a distributed environment or not) by executing a few simple commands. These commands automatically generate all the required files and directories, so that the user only has to optionally edit a straightforward configuration file (otherwise the default parameters will be used) before launching the job to the cluster.

[2] https://github.com/IntelCompH2020

2.5. High Performance Computing environment

The classification service has been designed to run in both: a single node mode, and in an HPC environment with multiple nodes. A model trained in an HPC environment follows the data-parallelism paradigm in which the training data is equally split between nodes and the model weights are replicated in each node. At each update, the communication between nodes allows the model to update each weight as if it was trained on the full batch of data. Since our software can run in 2 different types of clusters, the code has been designed to run in both types of hardware: Nvidia and AMD GPUs. In both clusters internet access is not available, therefore, all functions that require access to any online resources have been modified to look for the local resources instead.

3. DATA

This section describes the datasets used to train the ready-to-use classifiers that will be made available to the community. As it is explained, the provided service will allow the training of new models in other taxonomies provided that supervised data is available for the task.

3.1. PATSTAT

The Worldwide Patent Statistical Database (PATSTAT) from the European Patent Office (EPO) contains bibliographical and legal event patent data from more than 100 patent offices.

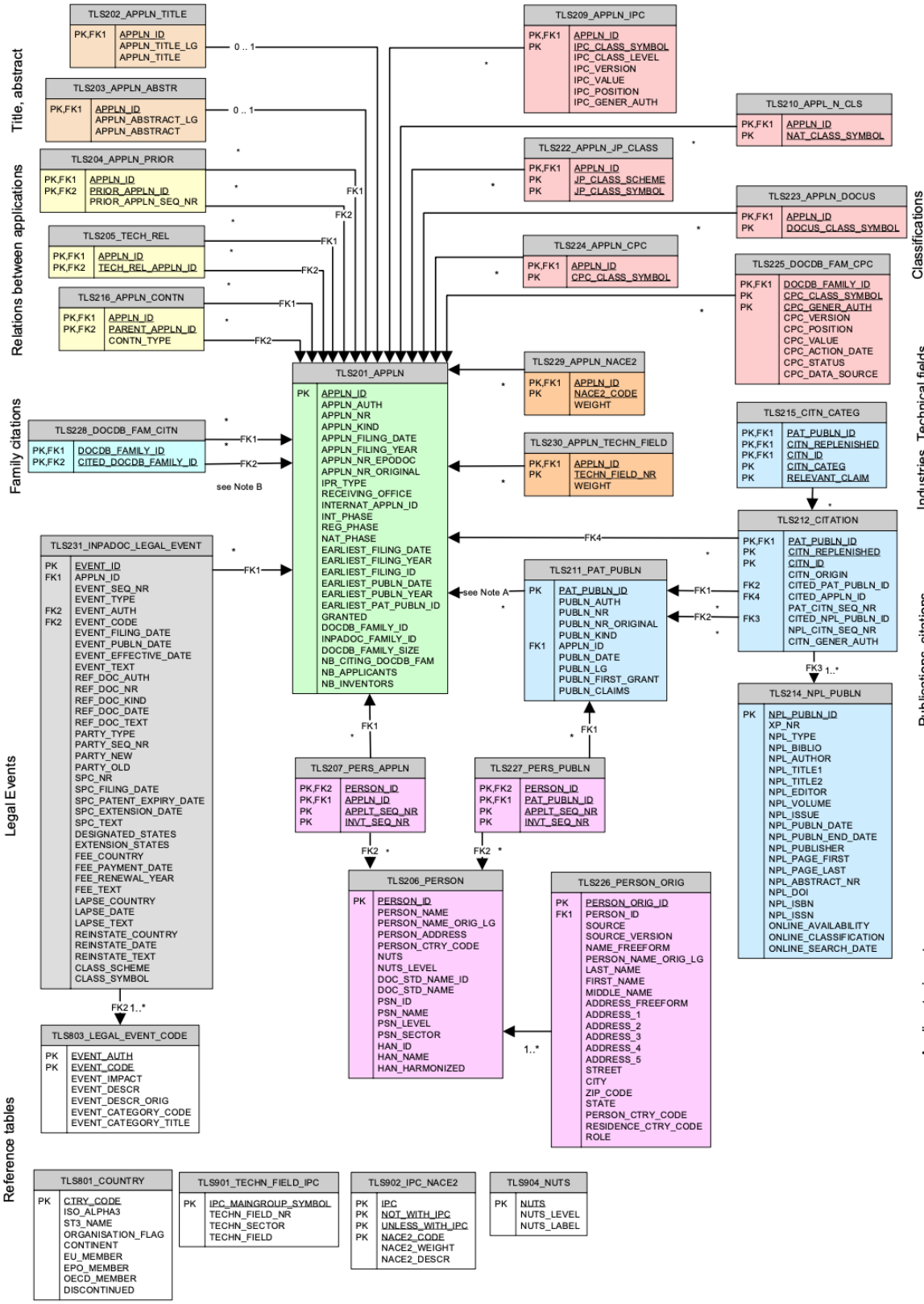
Instead of using the web-based interface, the data used for this project was gathered by querying the bulk dataset. The database's tables follow a relational database schema, with a central table (*tls201_appln*) that contains a separate entry for each application available in the PATSTAT database. It is important to note that several patents will belong to the same "family" if they cover the same or similar invention. For instance, if the same patent's invention is simultaneously filed in offices from several countries, every application will have a different application identifier (as it is used as the primary key in the database) but the same family identifier. EPO uses two types of families:

- DOCDB: *"All applications which are members of the same simple family do have the same priorities. The technical content of these family members is regarded as (almost) identical, so their publications are sometimes called "equivalent". "*
- INPANDOC: *"All applications which are members of the same extended family are directly or indirectly linked to the same root priority application. Usually the applications are related to the same technical invention, but their individual content may differ. "*

Every application belongs to exactly one DOCDB family (docdb_family_id) and one extended INPANDOC family (inpandoc_family_id). We filter out duplicated documents that belong to the same DOCDB/INPANDOC family, ensuring that there are no duplicates in our training dataset. Otherwise, we would have several training examples with almost the exact same text field, since the title and abstract content from patents within the same family is usually identical.

The figure on the next page displays a full overview of PATSTAT's database schema, with the attributes names contained inside the boxes that represent the different tables. To train the supervised text classifiers we are mostly interested in the title and abstract fields, which can be found in tables *tls202_appln_title* and *tls203_appln_abstr* respectively. All patents with one of these two fields empty were filtered out, as well as the ones that are written in a language other than English. In any case, note these decisions were taken to simplify the development of the service and validation through the training of a first model for patent classification. It is expected that at a later stage new classification models can be developed using either documents translated from other languages or patents for which only the abstract is available. Nevertheless, since the number of training instances for the initial model was already quite large we do not expect that a minor increase in the number of training examples would result in drastically improved results.

Figure 2. Logical model diagram of the PATSTAT Dataset[3]



[3] Source: [PATSTAT Data Catalog, Spring Edition 2022]

3.2. Microsoft Academic Graph (MAG)

Microsoft Academic is a project that leverages the cognitive power of machines to assist researchers in entity exploration of publications and knowledge discovery. Its main outcome is the so-called Microsoft Academic Graph (hereinafter MAG), which is basically a database with millions of records of scientific publications. The heterogeneous graph also contains metadata information such as the authors, affiliations, journals, fields of study and citation information. An entity disambiguation pipeline is used to do the mapping of those entities.

Microsoft uses a two-stage approach to automatically identify emerging fields of study from academic documents. It does so by first analyzing the document's vocabulary to identify field of study mentions (self-supervised sequence labeling task) and then a classifier maps those mentions to either existing or new fields of study. This allows the automatic discovery of new concepts without the need for pre-existing vocabulary seeds. The table below shows class examples at the different levels of the taxonomy, showing how concepts become more fine-grained at the lowest levels.

Table 1. MAG examples in different levels of the hierarchy.

L5	L4	L3	L2	L1	L0
Conv. Deep Belief Nets.	Deep Belief Network	Deep Learning	Artificial Neural Net.	Machine Learning	Computer Science
Reductase	Methionine Synthase	Methionine	Amino Acid	Biochemistry / Molecular Biology	Chemistry / Biology
Phosphatase	Phosphorylase Kinase	Glycogen Synthase	Glycogen	Biochemistry	Chemistry
	Fréchet Distribution	Generalized Extreme Value Distrib.	Extreme Value Theory	Statistics	Mathematics
Hermite's Problem	Hermite Spline	Spline Interpolation	Interpolation	Mathematical Analysis	Mathematics

Even though the documents are organized in a hierarchical structure composed of hundreds of thousands of scientific concepts divided in six differentiated levels, we focused exclusively on the second level of the hierarchy to train a zero-shot text classifier for the scientific domain. The reason is that deeper levels have an extremely high number of labels of lower quality since only the two top-most levels are manually curated while the rest are generated by a model, filtering reduces the dataset size making it more suitable for training a subsumption-based model.

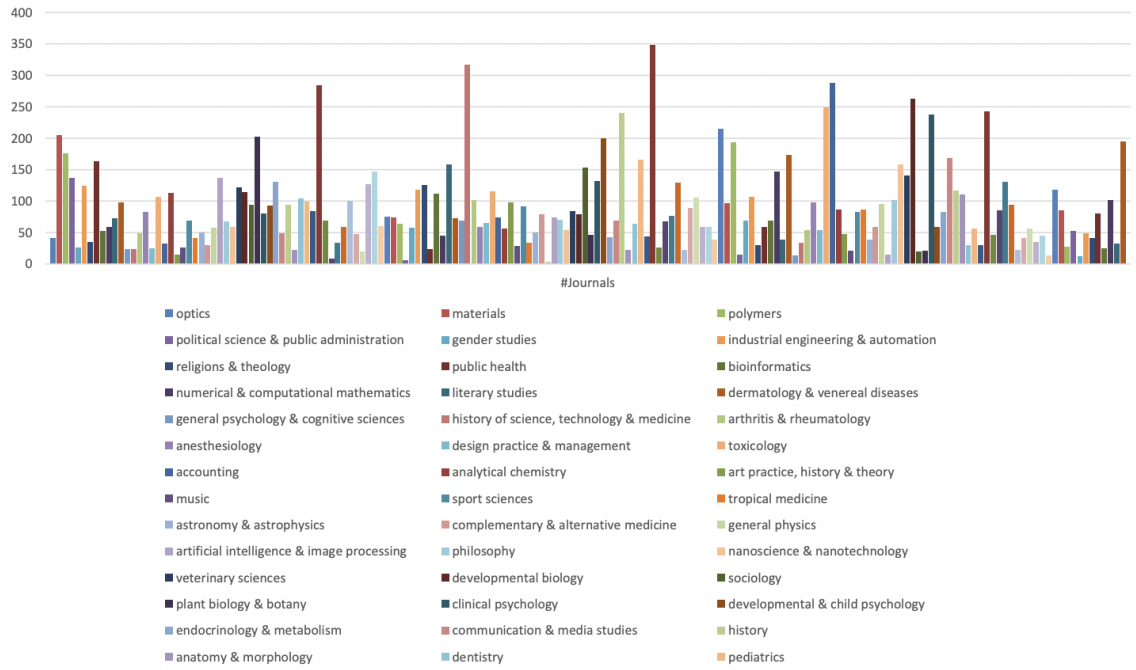
The knowledge graph was updated on a bi-weekly basis until the end of 2021, when Microsoft decided to stop providing this service. Our training data was retrieved from the latest dump.

3.3. Silver Field of Science Dataset

To create a Field of Science dataset containing publications along with metadata like titles and abstracts, we employ the *SCIENCEMETRIX*[4] Journal classification. This Journal classification provides journal names along with Field of Science categories. These Field of Science categories also exist in our Field of Science Taxonomy, which is described in section 4.3.

To create a comprehensive, large-scale, and clean dataset, we retrieve publications from Microsoft Academic Graph (*MAG*[5]) that are published in the Journals classified from *SCIENCEMETRIX*. *MAG* provides a wide range of publications. Figure 1 presents the number of Journals that *SCIENCEMETRIX* has classified to Field of Science categories.

Figure 3: Journals distribution per SCIENCEMETRIX FoS category



One can easily observe that by retrieving a certain amount of publications per journal, an unbalanced dataset will be created. We retrieve 500 publications per Journal and per FoS category. The unbalanced dataset created, describes real-world data, hence the evaluation splits (Train, Dev and Test sets) follow this unbalanced distribution. However, we would like the Train set to be balanced which means having equal number of publications per FoS category. To that end, we further sample *MAG* to obtain 10K train samples per FoS category. The final dataset statistics are presented in Table 1. For every publication, we also retrieve abstracts and additional metadata.

One limitation of the above-mentioned approach, is that the *SCIENCEMETRIX* classification is at the journal-level (introducing noise from multidisciplinary journals) and not at the publication-level. We try to mitigate this effect by filtering multidisciplinary (e.g *PLOS ONE*)

[4] https://science-metrix.com/
[5] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

journals and assume that the journal-level classification also represents the publication-level classification.

Table 2. Statistics of the Field of Science Dataset.

Statistics	Train Set	Development Set	Test Set	Total
Number of Instances	1.687.826	120.282	120.307	1.928.415

The silver Field of Science dataset can be accessed through this Link

3.4. Silver Sustainable Development Goals dataset

The General Assembly of the United Nations has adopted a global indicator framework for their Agenda for Sustainable Development. To measure the relevance of scientific articles to the Sustainable Development Goals (SDG) of the UN, an SDG classifier has been developed which classifies a given abstract of a scientific article to one or more SDG categories. A silver corpus of SDG-related articles has been developed using a controlled SDG vocabulary[6] to retrieve metadata of scientific articles found inCrossref. The controlled vocabulary contains for each SDG category multiple combinations of keyphrases that enable the immediate categorization of an abstract to the corresponding SDG category upon their appearance.

We created an ElasticSearch[7] cluster and indexed all CrossRef abstracts. Then we used all combinations of keyphrases as query terms and retrieved all abstracts containing a keyphrase combination, thus creating a collection of abstracts for each SDG category. Then for each collection of abstracts we applied keyphrase extraction using Textacy's[8] SGRank algorithm to extract additional keyphrases. A human curator reviewed the extracted keyphrases and augmented the controlled SDG vocabulary.

Finally using the augmented controlled SDG vocabulary we applied a second retrieval in CrossRef abstracts and MAG abstracts to form the final Silver SDG dataset. The Silver SDG dataset was used to train the SDG classifiers discussed in chapter SDG Classification.

4. Taxonomies

This section provides a general overview of the taxonomies used to train the text classifiers.

4.1. International Patent Classification (IPC)

IPC is a hierarchical system that provides an internationally uniform classification of patent documents. Intellectual Property Offices (IPOs) from over 100 countries use the IPC search tool for the effective retrieval of patent documents. As it is described in the official guide [1], it can be used as a basis for selective dissemination of information, to investigate the state-of-the-art in given fields of technology, and for the elaboration of industrial property statistics.

[6] https://zenodo.org/record/4118028#.YvY4GHZBxPb
[7] https://www.elastic.co/
[8] https://textacy.readthedocs.io/en/0.11.0/index.html

Since its release in 1968, it has been updated on a regular basis every few years. The number of classes has been changing over time, as well as the number of layers in the hierarchy. At the time of writing, the IPC taxonomy consists of 6 distinguishable levels, which we will refer to as section, class, subclass, group, main group and subgroup, respectively.

Each classification symbol is composed of:

- A letter to represent the section.
  - A: Human Necessities
  - B: Performing Operations, Transporting
  - C: Chemistry, Metallurgy
  - D: Textiles, Paper
  - E: Fixed Constructions
  - F: Mechanical Engineering, Lighting, Heating, Weapons
  - G: Physics
  - H: Electricity
- A two-digit number to represent the class.
- A letter to represent the subclass.
- A one-to-three digit number to represent the group.
- A two-digit number to represent the main group or subgroup.

The class is separated from the subclass by a double space, while the group numbers are separated by a backslash character. So, the full symbol would have the form “A01A 0/00” (with double space between “A01A” and “0/00”) . The short description associated with each symbol can be consulted using the interactive explorer from the WIPO IP portal.

The table below shows the number of classes that each section has at the different levels of the hierarchy (statistics provided by the official WIPO website).

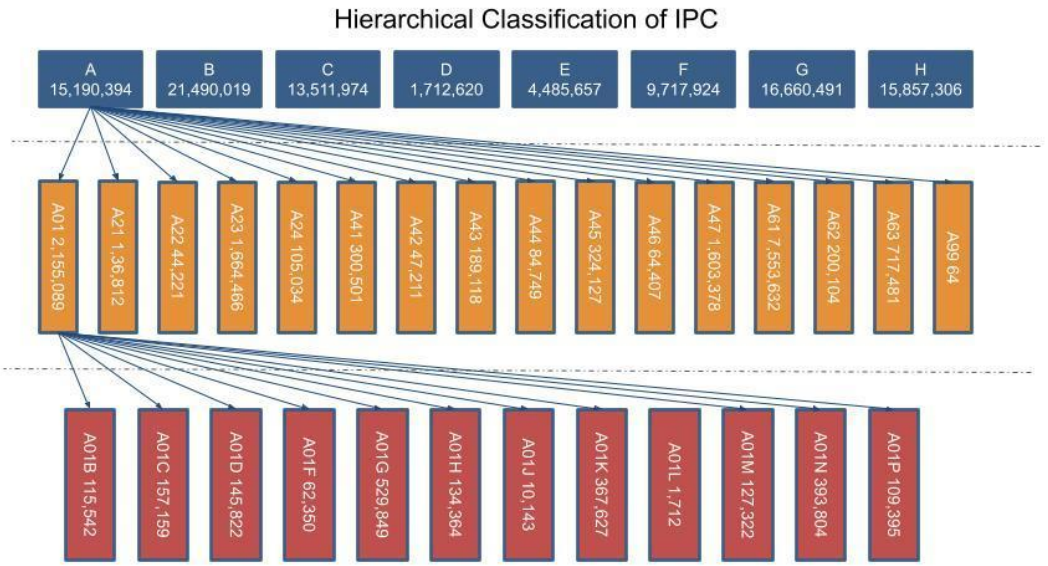
Figure 4: Number of categories in each level of the IPC hierarchy

Section	No. of classes	No. of subclasses	No. of main groups	No. of subgroups	Total no. of groups
A	16	84	1139	8485	9624
B	38	169	2001	15912	17913
C	21	87	1323	13644	14967
D	9	39	354	2895	3249
E	8	31	323	3122	3445
F	18	99	1105	8272	9477
G	15	87	741	8352	9093
H	6	51	559	9409	9968
Total	131	647	7545	70191	77736

Every document is assigned to at least one classification symbol, and more whenever it is possible to achieve a higher level of detail. The symbols can be considered of high quality as they are assigned by patent examiners that follow strict classification rules.

As shown below, the number of examples per section in the first level is relatively balanced, with some sections (i.e. E or D) well below the average. The imbalance is more pronounced at the second level though, as can be observed in section A’s classes whose numbers range from millions to a few hundred.

Figure 5: Number of examples per section of IPC level 1 - A - A01



During the examination of the available data, it was detected that only one class (namely G06V) had no examples at all. Interestingly enough, it was also seen that some of the classes available in the training data were not present in the current taxonomy. Since the number of documents assigned to these codes was very low in most cases, it was decided to remove them from the dataset. The codes and number of documents in each one of those classes are represented in the table below.

Table 3. Categories that are not present in the current taxonomy.

Subclass code	C12S	C13C	C13D	C13F	C13G	C13H	C13J	F24J
Num. Docs	2,641	6	15	14	5	1	2	60,737

4.2. Nomenclature of Economic Activities Version 2 (NACE2)

The NACE taxonomy is a European standard used to classify products and economic activities. This integrated classification system assigns different codes (henceforth “NACE codes”) to European industries based on their business activities.

The fact that it is compulsory to use NACE codes in the European Union makes it a reliable source of statistics regarding economic fields, having data that goes back to 1970. Note that some Member States use their own national nomenclature, but it must always be derived from NACE codes and preserve the same structure and hierarchy.

NACE has its own nomenclature for the codes, with a hierarchy that is four levels deep:

- Section: First character represented by an alphabetical letter.

- Division: The first two digits.
- Group: The third digit.
- Class: The fourth digit.

Table 4. NACE nomenclature.

Level	Sections	Range	Count
1	21	A - U	21
2	88	01 - 99	88
3	272	01.1 - 99.0	272
4	615	01.11 - 99.00	615

Note that additional levels might be added at the end by some countries or institutions, extending the total length of the code. Note also that each document can only belong to one and only one category.

The complete list of codes can be found on the European Commission's website, but for simplicity and obvious space reasons we only display here the first level of the taxonomy:

A. Agriculture, Forestry and Fishing
B. Mining and Quarrying
C. Manufacturing
D. Electricity, Gas, Steam and Air Conditioning Supply
E. Water Supply; Sewerage, Waste Management and Remediation Activities
F. Construction
G. Wholesale and Retail Trade; Repair of Motor Vehicles and Motorcycles
H. Transportation and Storage
I. Accommodation and Food Service Activities
J. Information and Communication
K. Financial and Insurance Activities
L. Real Estate Activities
M. Professional, Scientific and Technical Activities
N. Administrative and Support Service Activities
O. Public Administration and Defence; Compulsory Social Security
P. Education
Q. Human Health and Social Work Activities
R. Arts, Entertainment and Recreation
S. Other Service Activities
T. Activities of Households as Employers; Undifferentiated Goods and Services Producing Activities of Households for Own Use
U. Activities of Extraterritorial Organisations and Bodies

4.3. Field of Science Taxonomy (FoS)

The Field of Science Taxonomy is used as a classification scheme from the Field of Science (FoS) Classifier (Section 5.4) from now on referred to as SciNoBo, to automatically classify scientific publications to FoS categories at various levels of detail of the hierarchical structure.

More concretely, the aforementioned FoS classification scheme is underpinned by the OECD disciplines/fields of research and development (FORD) classification scheme, developed in the framework of the Frascati Manual[9] and used to classify R&D units and resources in broad (first level(L1), one-digit) and narrower (second level(L2), two-digit) knowledge domains based primarily on the R&D subject matter. To facilitate a more fine-grained analysis, we extend the OECD/FORD scheme by manually linking FoS categories of the *SCIENCEMETRIX*[10] classification scheme to OECD/FORD Level-2 categories, creating a hierarchical 3-layer taxonomy. Table 2 provides statistics of the FoS Taxonomy.

Table 5. Statistics of the Field of Science Taxonomy.

Levels of FoS	Number of Categories
Level 1	6
Level 2	42
Level 3	174

Table 3 provides somes examples of the Field of Science Taxonomy.

Table 6. Examples of the Field of Science Taxonomy.

Level 1	Level 2	Level 3
Natural Sciences	Physical Sciences	Optics
Social Sciences	Economics and Business	Economics
Engineering and Technology	Mechanical Engineering	Aerospace & Aeronautics

For a complete overview of the Field of Science Taxonomy, please refer to the OpenAIRE website.

[9] https://www.oecd.org/sti/inno/frascati-manual.htm
[10] SCIENCEMETRIX Classification provides a list of Journal Classifications with FoS categories. We get the lower level of this classification to manually link it with the OECD/FORD Level-2 categories.

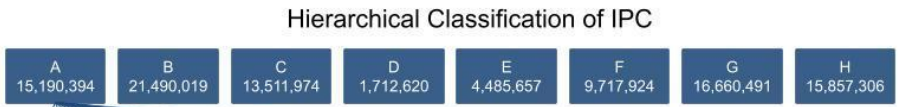
5. Trained Classifiers

5.1. Supervised Text Classifiers

5.1.1. IPC level 0

As explained in the previous section, the first level of the IPC taxonomy corresponds to the sections of the taxonomy. There are 8 differentiable sections:

Figure 6: Number of examples per section of IPC level 0

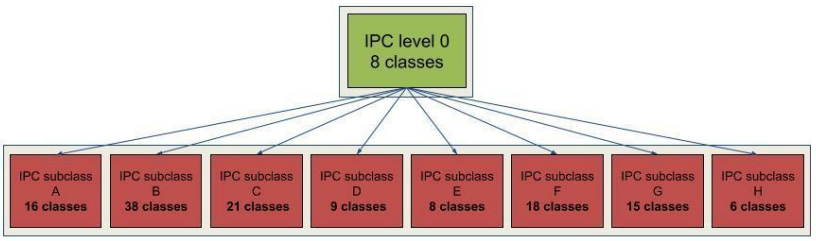


The number under each class indicates the number of patents available for training.

The RoBERTa-large model has been finetuned using only the title and the abstract of English patents. The classifier has used 90% of data for training, 5% for validation and 5% for test. The F1 score obtained after training is 0.8423.

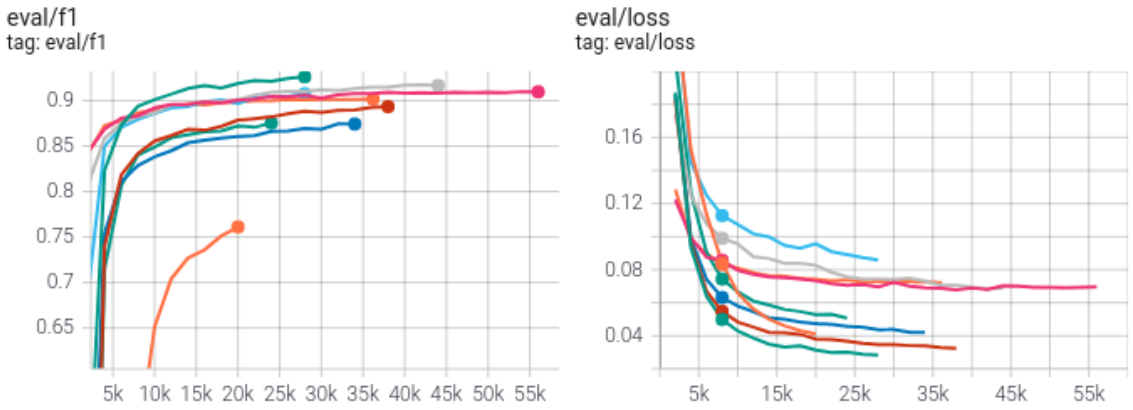
5.1.2. IPC level 1

Figure 7: Number of classes per section of IPC level 1



At the second level of the taxonomy, the classifiers achieve F1 scores of around 0.9. The underneath graphs, generated by *tensorboard*, show that this is the case for all models except the class B one, which is understandable since that class is by far the one with more subclasses to choose from (i.e. 38).

Figure 8: Tensorboard on IPC level1 models



Plot from tensorboard with training steps in the x-axis and eval f1 and loss in the y-axis. It can be seen that the orange F1 (that corresponds to the class B with 38 possible labels) is the worse performing model, this can be explained by the high number of possible labels.

5.1.3. NACE2 level 0

In a similar manner, a classifier was trained on the top level of the NACE2 taxonomy. In this case, the F1 score achieved is 0.7953.

5.1.4. NACE2 level 1

The scores reported for the 7 classifiers trained at this level. Out of the 26 level 0 classes we only have trained 7 classifier because the rest do not have more subclasses in our training data (they are the end of the hierarchy). The minimum F1 obtained is 0.8506, and the maximum 0.9864.

5.2. Zero-Shot Text Classifier

Zero-shot text classification is a widely studied task that deals with the lack of annotated data, using models that are able to classify text into classes that have never been seen during training. The main advantage is that it removes the need for time-consuming annotation processes, but also reduces the computational cost of having to fine-tune a different model for each application. In the case of the Intelcomp project, a zero-shot text classifier has been trained for scenarios where no training data is available. In any other case, the classifiers presented in the previous section (or the ones that will be trained in the future) should be the preferred choice, as traditional fine-tuning with annotated data leads to higher performance.

Following the most common approach in the literature, the model was trained on the Recognizing Textual Entailment (RTE) task. The training data consists of a modified version of articles and fields of study extracted from the Microsoft Academic Graph (MAG). The abstract is given to the model as the premise of the entailment task, while the hypothesis is built by concatenating a specifically designed prompt and the field of study. For instance, the prompt could have the following format: “This example is about {LABEL}”. In such case, for every given example, the model would receive two sequences separated by a special token:

- Seq. 1: Content of the document (could be the title, abstract or their concatenation).
- Seq 2: The sentence “This example is about {LABEL}”, with the placeholder being replaced by the actual class.

Then, during fine-tuning, the models learns to discern whether the hypothesis (sequence 2) is entailed by the premise (sequence 1) or not, enabling inference with new classes by computing the entailment score between the input text to be classified and the candidate classes given as options. The final prediction of the model will be the class associated with the higher score, or all the classes above a certain threshold in a multi-class setup. The final model, which has a RoBERTa-large architecture, achieves a state-of-the-art F1 score in the scientific domain and competitive results in other areas.

This work led to the publication of a scientific paper at the upcoming EMNLP conference, which is undergoing a peer-review process at the time of writing.

5.3. Sustainable Development Goals (SDG) Classifier

We used the Silver Sustainable Development Goals dataset to train an SDG classifier for abstracts of scientific articles. We trained two deep learning models based on a distilled version of BERT. We used a pre-trained version of a distilled BERT model and further finetuned a Multilayer Perceptron on top of the BERT model using the silver SDG corpus. During training we excluded data from underrepresented categories. We trained two models, one taking into account the vector representation of each token of the input through an attention mechanism and one model that takes into account the vector representation of the entire input sequence. As the deep learning models tend to be sensitive to key phrases found in the controlled vocabulary and several SDG categories are underrepresented in the corpus, a guided LDA topic model was also trained using the key phrases found in the vocabulary. During training, a set of key phrases is assigned as a seed to each topic and the topic is labeled with the corresponding SDG category. Guided LDA is thus given a fixed prior probability distribution of word occurrence for each topic. Then LDA is further tuned with the entire silver SDG corpus and additional randomly selected articles from CrossRef.

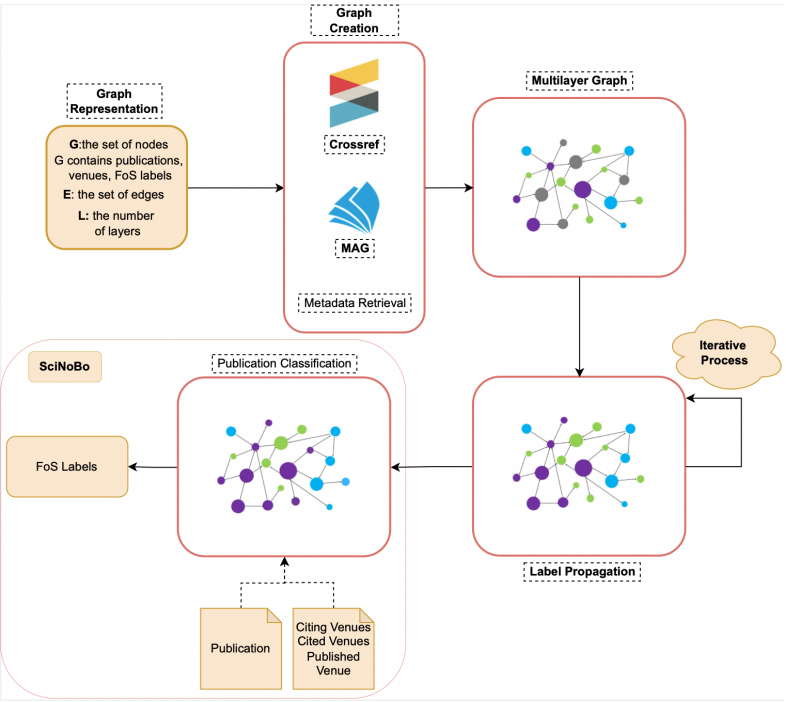
Overall the classifier comprises two deep learning models and a guided LDA model combined via a voting strategy to classify an unseen article abstract. When an unseen scientific abstract is processed if a phrase of the controlled vocabulary is present the abstract is categorized into the corresponding SDG class. Then the guided LDA assigns a score to each topic and their corresponding SDG categories and the two deep learning models compute a probability distribution across the trained SDG categories. When the computed scores exceed predefined tunable thresholds the abstract is assigned its SDG categories.

5.4. Field of Science (FoS) Classifier - SciNoBo

SciNoBo classifies publications to one or more FoS (Section 4.3) and is based on the assumption that a publication[11] mostly cites thematically related publications. We bridge venues (journals/conferences) and publications by constructing a multilayer network (graph) in which venues are represented by nodes, and venue-venue edges reflect citing-cited relationships in their respective publications. SciNoBo classifies through the publishing venues of the publications it references (out-citations) and the publishing venues of the publications it gets cited by (in-citations). Therefore, SciNoBo classifies publications with minimal metadata utilizing only journal or conference names as well as citing information. Figure 2 illustrates the steps followed to create SciNoBo. Each step of the approach is covered in the following subsections.

[11] We use the term "publication" to refer to all peer-review research works published in journals or conferences.

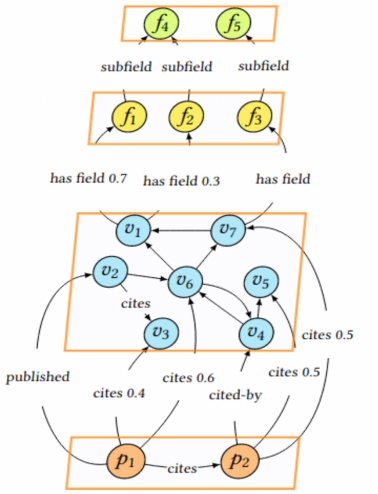
Figure 9: Illustration of SciNoBo. After we define our graph in Graph Representation, we retrieve metadata and construct it. The result is a Multilayer Graph and after Label Propagation, we can input a publication along with the required metadata to retrieve the FoS labels.



5.4.1. Graph Representation

SciNoBo unifies multiple types of relationships (edges) between entities as well as multiple types of entities under a common framework of operations represented as a multilayer network[12] (Figure 3 provides a representation of the multilayer network).

Figure 10: Schematic representation of the multilayer network.



[12] Multilayer networks are data structures used to model complex interactions, ranging from Biomedicine , to Social Network Analysis.

Consider a multilayer network $G = (V, E, L)$. V represents the nodes of the graph which can be publications (P_i), venues (V_i);conferences or journals and FoS categories (F_i) as defined by the FoS Taxonomy (Section 4.3). E represents the set of edges (links) between the nodes and L is the set of layers capturing different types of relationships between nodes. Since the network has multiple layers, each edge belongs to one layer ($l \in L$) and has a weight of $w \in R^+$. We can represent all edges in the network by utilizing 4-tuples, e.g. $e_i = \{(u, v, l, w);\ u, v \in V,\ l \in L, w \in R^+\}$ with e_i being a certain edge. Edges in layer l represent a particular type of connection among nodes, and two nodes u, v might be connected by edges in multiple layers.

We formulate the task of scientific field classification as a link prediction problem in the multilayer network. The goal is to predict edges between publication and scientific-field nodes.

The edges at different layers l correspond to different interpretations. More concretely, an edge between publications (p_i, p_j) at l_1 means that p_i cites p_j. An edge at layer l_2 connects a publication to its publishing venues(s). We also define edges at layer l_3: $\{(u, v, l, w);\ u, v \in V,\ l \in L, w \in R^+\}$ where w is the number of publications which have been published in venue u and cite (reference) publications published in v. The weight of an edge (v, f) at l_4 corresponds to how thematically relevant the publications published in v are to the scientific field f (FoS). Finally, l_5 + layers represent hierarchical relationships among labels.

5.4.2. Graph Creation

SciNoBo network was populated by exploiting Crossref[13] and Microsoft Academic Graph (MAG)[14]. Crossref contains more than 120 million publications and MAG contains approximately 250 million records. We retrieve all the publications that were published between 2016−2021, along with their references and their citations when available. We confine the references in a 10 year window. For every publication, the publishing venue is contained in the metadata. However this is not the case for the references and citations. As a result, for every publication we query its references and citations in Crossref/MAG (by taking the union of the metadata) and we retrieve the original metadata of the reference or the citation. Inherently now we can create venue-venue edges (edges at layer l_3) as described in 5.4.1. The weight of a venue-venue edge is the amount of times a venue has referenced or being cited by another venue. Post graph creation, we normalize the weights of each node's outgoing edges to sum up to 1. The normalized weight of a venue-venue edge (u, v) can be roughly interpreted as the probability of a publication published in u to cite a publication published in v.

[13] https://www.crossref.org/
[14] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

After the Graph Creation step is finished we get a Multilayer Graph as depicted in Figure. 2.

5.4.3. Label Propagation

As mentioned in section 4.3, we utilize *SCIENCEMETRIX* classification to manually link the Level 2 Field of Science categories with the *SCIENCEMETRIX* ones thus creating the hierarchical FoS classification scheme. Furthermore, we utilize *SCIENCEMETRIX's* journal classification (FoS categories), by mapping its journals to the nodes of the multilayer network of SciNoBo (recall that the nodes can be venues). This mapping represents l_4 relationships, which are utilized to classify publications in FoS labels. Initially a small portion of venues have an FoS in Level 2 and Level 3. By utilizing Label Propagation, we aim to increase the venue label coverage.

The benefit of incorporating venues into the multilayer network, is that starting from a small set of seeds (venues with FoS categories), we can propagate the information to the rest of the network. The intuition behind this approach is that a venue is more likely to express the FoS category of its most referenced venues. However, we do not consider these venue-FoS assignments to be ground-truth and we dynamically re-evaluate them during Label Propagation. By taking into account the network of venue-venue relationships, we enrich the initial FoS journal classifications described in this section by inferring additional venue-FoS relationships. Consequently, previously single-labeled classifications (of venues) may become multi-labeled after a few rounds of label propagation.

5.4.4. Publication Classification

Publication-classification uses the same label propagation mechanism as the one presented in subsection 5.4.3. Assume that each publication is represented by a unique node in the SciNoBo network. The goal is to connect each publication node to one or more FoS nodes. We have already discussed how venue-FoS relationships (l_4) can be established in the subsections 5.4.2 and 5.4.3.

There exist multiple ways to back-propagate information from the venue level to the publication level depending on the available metadata and are listed below:

1. based on the published venue (namely *Published-by*)
2. based on the referenced/cited venues (namely *References*)
3. based on the referenced (cited) and citing venues (namely *References+Citations*)

Published-by: Given a publication p and the set of distinct venues (nodes) it has been published in (most of the times equal to 1), we draw edges of equal weight from p to the venues (nodes). As a result each published venue only contributes the weight it has with its FoS categories (edges at l_4). The scores per FoS are aggregated and ranked according to their total weights. The publication is finally classified to the top T FoS, where T might be fixed or be equal to the number of weights that exceed a user-defined threshold.

References: Given a publication p and the set of distinct venues it references $K = \{v_1, v_2, ..., v_k\}$ we draw edges between p and the venues with weight $w_{p,v_i} = (number\ of\ referenced\ publications\ published\ at\ v_i/k)$. Similar to the *published-by* approach the weights are aggregated and the publications are assigned to the top T FoS.

References+Citations: This approach is identical to the *References* one. However, we also take into account the venues that cite publication p (*cited-by* edges in Figure. 3 if and when available. A methodology originally proposed in the context of one particular field might eventually prove groundbreaking in a completely different field. By incorporating citation venues, SciNoBo captures cross-domain FoS that would otherwise be missed.

The analysis of SciNobo (Field of Science classifier), the Field of Science Taxonomy (Section 4.3) and the Silver Field of Science Dataset (section 3.3) was adapted from the paper: **SciNoBo: A Hierarchical Multi-Label Classifier of Scientific Publications**. For more analysis and experimental results please refer to the paper, which was published in Sci-K 2022, co-organized with The Web Conference 2022.

6. Conclusion

This document presented the classification service that will be part of the AI workflows required to obtain valuable insights from unstructured data. The ultimate goal is to support evidence-based policymaking, which is one of the main missions of the Intelcomp project. The presented text classifiers are an additional contribution that brings the Intelcomp project one step closer to such goal, ensuring that the tool does not become obsolete in the future thanks to a model trainer that can be easily used out-of-the-box.

REFERENCES

[1] World Intellectual Property Organization. (2018). Guide to the International Patent Classification.

[2] Cohan, A., Feldman, S., Beltagy, I., Downey, D., & Weld, D. S. (2020). Specter: Document-level representation learning using citation-informed transformers. arXiv preprint arXiv:2004.07180.

[3] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. ArXiv, abs/1910.03771.

[4] 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.

ANNEX I: HOW TO USE SUPERVISED CLASSIFICATION MODEL TRAINER

This section provides the steps to follow in order to train classifiers on new taxonomies using the trainer of supervised classifiers:

- Clone the taxonomical-classification repository:

```
git clone git@github.com:IntelCompH2020/taxonomical-classification.git
```

- From the newly created directory, install the required libraries in a virtual environment.

```
bash setup_environment.sh
```

- Create a working directory for the new taxonomy:

```
bash new_taxonomy.sh taxonomy_name=<TAXONOMY_NAME_HERE>
```

- Download the model to be finetuned using the scripts from the `./models` directory, or alternatively manually placing them in there.

```
bash models/download_roberta_large.sh
```

- Go to the taxonomy directory and edit the configuration file, where you can set hyperparameter values as well as HPC specifications (number of nodes, number of CPUs…). If you want to run the code locally, simply set the "hpc" variable to `false` and ignore the rest.

```
vim hyperparameters.config.sh
```

- Then you can generate the scripts that will be used to launch the job.

```
bash generate_run.sh
```

- And finally, you can either run the code locally:

```
bash run.sh train_files=X dev_files=X test_files=X text_column=X label_column=X
```

  Or send a job to BSC's facilities:

```
sbatch launcher.sh train_files=X dev_files=X test_files=X text_column=X label_column=X
```

The step by step instructions as well as the source code can be found in the GitHub repository[15].

[15] https://github.com/IntelCompH2020/taxonomical-classification