BibHelioTech internship report

Axel Dablanc, Vincent Génot, Camille de Salabert, Sabine Barreaux, Pascal Cuxac, Nicolas Dufourg, Nicolas Aunai, Williams Exbrayat

Report author:

Dablanc Axel

IUT tutor :

Cabanac Guillaume

Internship tutor :

Génot Vincent

Internship dates :

Du 28/03 au 24/07 2022

INTERNSHIP REPORT

BibHelioTech

I would like to thank all the people who contributed to the creation and success of my internship and helped me write this report.

First of all, I'd like to thank my teacher, Mr Guillaume Cabanac from IUT III in Toulouse, who helped me a lot in my search for an internship and enabled me to apply to this laboratory. His attentiveness and contacts enabled me to find this internship, which was totally in line with the realization of my dream.

I'd also like to thank my internship supervisor, Mr. Vincent Génot, astrophysics researcher at IRAP (INSTITUT DE RECHERCHE EN ASTROPHYSIQUE ET PLANÉTOLOGIE), for his warm welcome, the time we spent together and the daily sharing of his expertise. Thanks to his confidence, I was able to fully achieve my goals. He was a great help in setting up the metrics and constructing the Excel data.

Table of contents

Illustrations

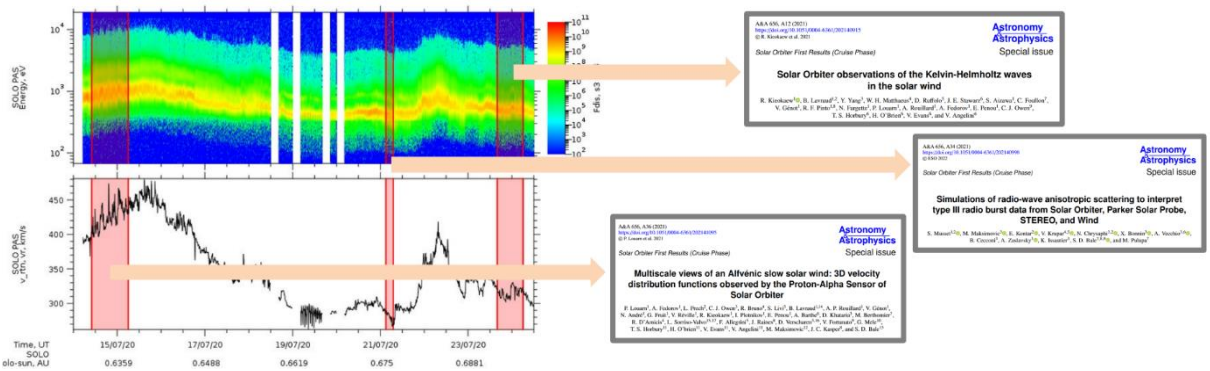
Introduction

In the field of astrophysical research, when researchers study physical phenomena, they use various satellites to store their data, which will only be transcribed in the researchers' publications at the end of their studies.

Because of this lack of a common database, it is very difficult to make the link between an observation made "x" time ago by researcher "X" and another observation similar to the previous one made "y" time ago by researcher "Y", unless one has read and remembers all the publications.

AMDA is a web tool that lists measurements made by various satellites and plots them against time.

Vincent Génot, researcher and senior lecturer at IRAP, wants to develop a computer program to extract from researchers' publications their DOI (unique identifier), the name of the missions (the satellites), their instruments, the region of space in which the measurements were taken, and the time interval of these measurements in the form of start time - end time, respecting the ISO 8601 format such as: YYYY-MM-DDTHH:MM:SS.zzz. All this was done with the aim of integrating this data into AMDA, so as to be able to easily and visually identify the time intervals covered by an article, and to be able to consult this article via a clickable link, as shown below.



[Fig.1 – Final customer requirement]

Company and professional context

Every year, IRAP welcomes around a hundred trainees from bachelor's, master's and engineering schools, as well as doctoral students, the vast majority in physics, but also in computer science.

Due to a lack of space for the trainees, I worked in a room with five other trainees, 3 from ISAE supaero, and 2 university physics students. My supervisor's office was a little further down the corridor.

During my internship, I had 3 different computers at my disposal, so as to be able to get the machine best suited to my needs. First of all, a low-powered computer, followed by a more powerful but slower remote machine called EDE2, and then, at the end of the course, a fast and powerful machine.

Analysis of the work to be carried out, methods and tools used

Textual extraction

Some fifty PDF files containing articles written in English by researchers were made available to me for processing.

The first step in processing the files is to have them read word for word by a machine, in order to process and locate the various entities required for the project. Among these entities, it is necessary to be able to identify the names of the missions, corresponding in most cases to the names of satellites; the names of the instruments on board these satellites which carried out the measurement(s); the time interval during which the measurement(s) was (were) carried out; the region of space in which the mission operated; and the DOI of the article, corresponding to a unique identifier serving as the article's identity number.

To illustrate this point, here's an example of an article excerpt featuring color-annotated entities.

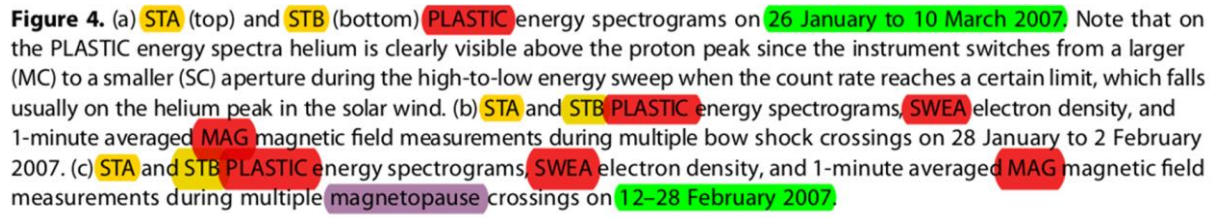


Figure 4. (a) STA (top) and STB (bottom) PLASTIC energy spectrograms on 26 January to 10 March 2007. Note that on the PLASTIC energy spectra helium is clearly visible above the proton peak since the instrument switches from a larger (MC) to a smaller (SC) aperture during the high-to-low energy sweep when the count rate reaches a certain limit, which falls usually on the helium peak in the solar wind. (b) STA and STB PLASTIC energy spectrograms, SWEA electron density, and 1-minute averaged MAG magnetic field measurements during multiple bow shock crossings on 28 January to 2 February 2007. (c) STA and STB PLASTIC energy spectrograms, SWEA electron density, and 1-minute averaged MAG magnetic field measurements during multiple magnetopause crossings on 12–28 February 2007.

[Fig.2 – Extract from a research publication]

durations, missions name, instruments, region

There are several possible methods for extracting text from PDF files. These include:

- The most basic "PDF to TXT" method, which simply reads the characters encoded in the PDF file. It's fast and resource-efficient, but it can't read characters embedded in images, or if the PDF file is itself derived from a scanned document; this is known as PDF-images..

- The "PDF to XML" method is relatively similar to PDF to TXT; the difference is that it structures the text in tags so that extracts from the article, such as the title, a particular paragraph, the conclusion, etc., can be easily located.
  - Below is an illustration of an XML file:

[Fig.3 – Extract from XML file]

This method is slower, more resource-hungry and still doesn't extract text from images. On the other hand, it has the advantage of being able to easily target the passages that interest us for further processing, thanks to the.

- The OCR (Optical Character Recognition) method, on the other hand, uses AI (Artificial Intelligence) to visually recognize the shape of characters in an image.

  Optical Character Recognition (OCR) uses AI (Artificial Intelligence) to visually recognize the shape of characters in an image, in order to transcribe all the text (including text embedded in an image) contained in an article into a TXT file. It is very slow and resource-intensive, but is a reliable method for guaranteeing the extraction of all textual information from a document.

Entity recognition

In order to be able to extract the various entities from the articles, the program must be able to recognize them, but also be able to differentiate between them. In other words, among all the words in an article, the program must be able to recognize those that are useful to our project, but also among all these useful words, be able to differentiate which are time intervals, satellites, instruments, space regions and which is the DOI of the article.

Temporal entities

It's important to remember that a time can be written in different ways. For example, for a date, January 15, 2022 can be written textually as written just now, or 15/01/2022, or 01-15-2022, or 2022-01-15, etc. In order to be able to recognize and extract temporal entities, it is essential to be able to identify a date or duration in all or, failing that, a maximum of these different formats.

Also, not all the dates contained in an article concern measurements. An article also contains its date of receipt, validation and publication, as well as the dates of bibliographic references.

In addition, as part of this project, times must be formatted in ISO 8601 format in the final results.

So, in order to recognize and extract temporal entities, it is necessary to be aware of tools capable of doing this, if they exist. If not, create these tools. But also filter out unwanted dates.

Named entities

In order to recognize and extract the various named entities we need, it is necessary to find or design a tool capable of detecting in text the name of a satellite or instrument or space region. What's more, the entities must be correctly linked to each other. For example, if 4 satellites and 15 measurements are found, the program must be able to understand that the first satellite measured measurements 1,2,3, the second, measurements 4 and 9, and so on. Or the first satellite is cited anecdotally, and only the other 3 have taken measurements in the article. So many cases that need to be dealt with.

We also need to take into account ambiguities between entities and other words that have the same name, but are not what we're looking for in the context of the sentence. For example, if one of the satellites is called "wind", it can easily be confused with the word "solar wind", which in this case makes no mention of the satellite in question.

It is also necessary to have a way of managing the acronyms of these entities. These articles are written by hundreds of different researchers who write in different ways. For example, let's take the case of one particular instrument, "SWIA". While one researcher may write it as "Solar Wind Ion Analyzer", another may choose to write it in the short form "SWIA".

Finally, various methods are available for entity recognition.

- The first is to use an AI capable of recognizing the type of each word in order to target the desired ones. This is known as NER (Named Entity Recognition). It is based on POS tagging (Part Of Speech) capable of detecting the gender of the word (illustration below).

[Fig.4 – Illustrion POS-Tagging]

As a result, the NER will be able to differentiate, for example, "I eat an apple" from "I work for apple". In which case, the former will be tangued, for example, "food" and the latter "enterprise".

NER is a very slow, resource-intensive method, and even extremely resource-intensive when it comes to training the AI in new tags. However, it does have the advantage of being able to correctly tag words that are not part of its training, provided it has had good training beforehand.

- The second method is to build up a database of all the entity names we need (illustrated below), and search for them by comparison in the text. This is a much faster and less resource-intensive method, although it does require a human to build the database with all the names it needs.

Cluster-Rumba	Cluster 1	Cluster-1	CL1	Cluster Rumba	Cluster1
Cluster-Salsa	Cluster 2	Cluster-2	CL2	Cluster Salsa	Cluster2
Cluster-Samba	Cluster 3	Cluster-3	CL3	Cluster Samba	Cluster3
Cluster-Tango	Cluster 4	Cluster-4	CL4	Cluster Tango	Cluster4

[Fig.5 – Extract from Excel file]

Finally, you need to be able to link the right measurement taken by the right satellite to the instruments belonging to the right satellites in the context of the article.

DOI

The DOI of an article has the following form: 10.1029/2006GL027188.

In order to find the DOI of the article in question, you either need to find a tool already capable of doing this, or to design a program capable of doing it, thanks in particular to regex (regular expression), which is a tool capable of detecting a word or any other textual value that resembles a given pattern. For example, the pattern [0-9]{4} means "I'm looking for a word made up of 4 digits from 0 to 9", or finding the DOI of an article from its title by querying a website via its APIs. Simply put, an API is a tool that enables a computer program to request information from a website.

Writing the final result

Finally, the results must be written in a text file readable by the AMDA tool, in the so-called HPEvent format.

Analysis and methodology: approach adopted, planning, choices made, tools used, errors corrected, etc.

Textual extraction

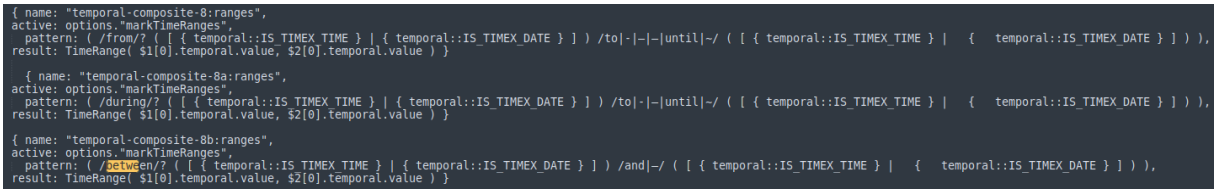
A series of text extraction tests were carried out using the different methods mentioned above. In view of the results obtained, the ocerization method (python package tesseract) was chosen for text extraction. Whether PDF to TXT (PyMuPDF python package) or PDF to XML (GROBID python package), in both cases the program was unable to extract all the text. Paragraph halves and even entire tables were missing. What's more, although OCR is slow and costly, the fact that it can reliably extract the entire text of an article, including text in images, is a major added value for the project. This means that the program can process the entire text of an article.

Entity recognition

Temporal entities

In order to extract and format time values in ISO 8601, my search for tools led me to select two candidates. The python packages dateparser and SUTime.

After a series of tests to evaluate the potential and capabilities of each, SUTime was finally selected for the project. Indeed, dateparser was capable of recognizing very few temporal values compared with SUTime. What's more, SUTime has the enormous advantage of using AI via the CoreNLP python package, which with training enables us to improve results, and it also gives us the possibility of adding rules (in regex (illustration below)) to enable it to recognize more temporal values, but also to refine the result of these.



[Fig.6 – Extract from the SUTime rules file]

In order to evaluate the results of SUTime in its basic version, I carried out a complete processing of all the articles, and evaluated for each of the temporal values, if this one is a value of interest to us (not a date in the references for example), the output result of SUTime as follows:

- Satisfactory result: date and/or time are correct, as is ISO 8601 formatting.
- Partial result: date and/or time are incomplete (no year or day (e.g. 2022-01T15:25:32 or 06-1501T15:25:32 or 2022-01-15T15:25 (no seconds specified)) or date without time (2022-01-15) or time without date (T15:25:32).
- Incorrect result: wrong year, wrong time, wrong day, etc.

The following diagram shows the percentages of each of these results.

[Fig.7 – Evaluating the basic performance of SUTime]

Although SUTime is much more powerful than dateparser, the results were still far from satisfactory enough to meet the needs of Vincent Génot and his project.

Since I have the possibility of adding rules to SUTime to improve these results, I thought it would be useful and necessary to look into the matter, but also to analyze the source of these errors in order to establish areas for improvement.

Thus, the main sources of error leading to partial or incorrect results were due to:

- Dates that are too important to exclude from processing, such as article reception, acceptance and publication dates. But also dates contained in references.

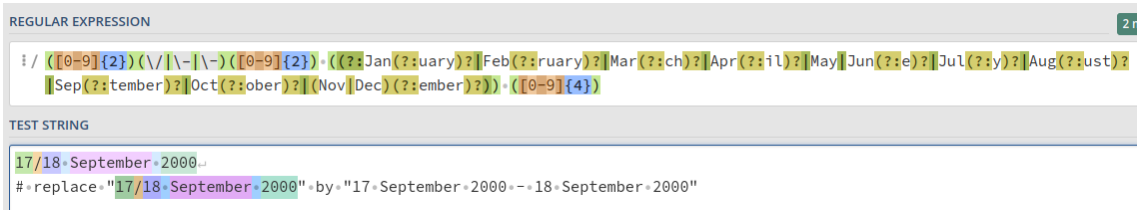
- Dates formatted in such a way as not to render all the information in the result, as illustrated below.:


```
"21 January 2016 01:06:41.10-01:06:52.04"
[
    {
        "end": 39,
        "start": 0,
        "text": "21 January 2016 01:06:41.10\u201301:06:52",
        "type": "DURATION",
        "value": {
            "begin": "2016-01-21T01:06:41",
            "end": "T01:06:52"
        }
    }
]
```


[Fig.8 – Example of SUTime output]

- In this example, we can see that the date is correctly formatted in the "begin", but is not carried over to the "end".
- Formats that are purely incomprehensible to SUTime, such as the HHmm format or the DOY (Day Of Year) format, which means, for example, that February 3 of 2017 is day 34 of 2017.

In this case, the idea was to use regexes to locate all these problematic shapes and transform them so as not to alter their value, while at the same time making them more comprehensible to SUTime. One of these transformations is illustrated below:



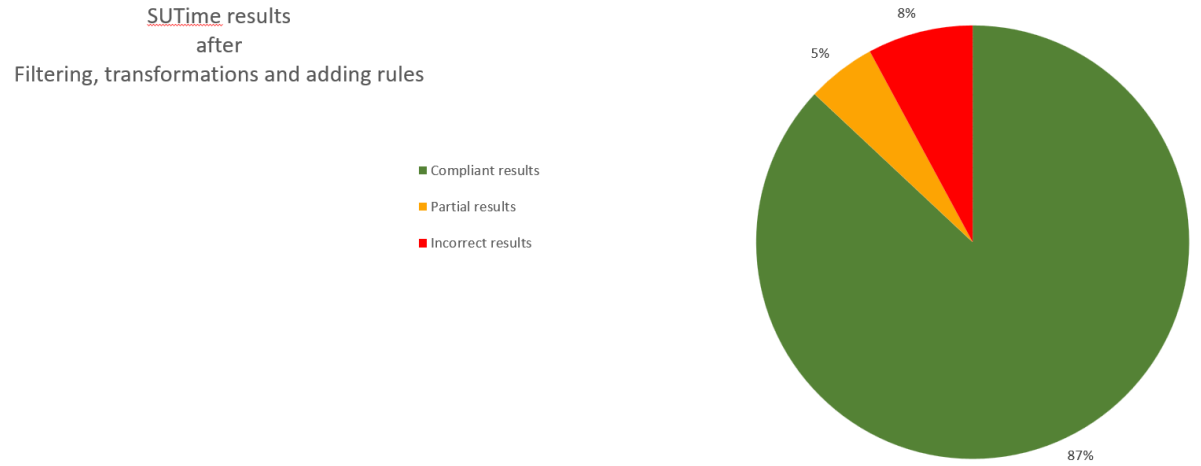
[Fig.9 – Example of date transformation using the RegEx method]

The problematic formatting is detected by the regexes, then the missing years/months/days/hours/... in the right or left term, but not in the other term, are copied and pasted by substitution.

Once the filters needed to solve this problem had been developed, the following errors were still present:

- Years sometimes auto-completed with the year of program execution by SUTime when this was missing or completed with "XXXX"..
  - ✓ Corrected by searching for the year closest to the missing year.
- Missing days, because not specified in the source text ("in september 2013" results in 2013-09 (no day)).
  - ✓ Corrected by assigning the day closest to the missing one, provided that this day is in the same month of the same year.
- Times of different lengths.
  - ✓ Corrected by harmonizing the tenses. If they end in hours, they are completed by begin: HH:00:00.000, end : HH:59:59.999, and so on, ending with minutes, seconds...

Solutions were then implemented to correct the various problem cases. Following this, we carried out a new evaluation of the time-value recognition results after the various modifications added to the SUTime rules and to the SUTime program.



[Fig.10 – Evaluation of modified SUTime performance]

As we can see, these multiple modifications have enabled us to significantly improve our program in terms of time entity recognition.

Named entities

After a great deal of research into methods for recognizing named entities, I finally opted for a database and in-text comparison method. Firstly, because I didn't have a suitable machine at my disposal to carry out satisfactory training on AI. But also because it seemed fair to me to judge that, given that the quality of entity recognition results between the NER (AI) method and by textual comparison would be relatively equivalent; knowing that the AI method would be very slow compared to the other method; that the AI method would only be beneficial if it were then able to understand words in context, as well as a human, in order to correctly link the right satellite to the right measurement in the right spatial region. However, my research has shown that such an AI would require far greater computing power than I have at my disposal, and that such an AI does not yet exist in any case.

With the help and support of Vincent Génot, we set up a database listing all the satellites with their names, name variants and nicknames; the instruments with their long and short names (acronyms), and the space regions.

This database takes the form of a simple Excel file, with one sheet for each of the different entity types. This file was completed using information retrieved from AMDA's internal data; the hpde data maintained by the University of California in collaboration with NASA; and the nssdc, a data source managed by NASA.

Below are illustrations of the different sheets in the Excel file :

- Satellites

BepiColombo	BepiC	Bepi	Bepicolombo	Mio	MPO
CARISMA					
CHAMP					
CLIMAX					
CNOFS					
CORONAS-F					
CRRES					
Cassini					
Cluster-Rumba	Cluster 1	Cluster-1	CL1	Cluster Rumba	
Cluster-Salsa	Cluster 2	Cluster-2	CL2	Cluster Salsa	
Cluster-Samba	Cluster 3	Cluster-3	CL3	Cluster Samba	
Cluster-Tango	Cluster 4	Cluster-4	CL4	Cluster Tango	
Cluster					

[Fig.11 – Satellite Excel spreadsheet extract]

- Instruments

BepiColombo	{'MAG':'Magnetc	{'ENA':'Energetic	{'MEA1':'Mercury	{'MEA1':'Mercury	{'MGF':'Mercury	{'MIA':'Mercury I	{'MSA':'Mass Spe	{'MEA':'Mercury Electron Analyzer'}
BepiC	{'MAG':'Magnetc	{'ENA':'Energetic	{'MEA1':'Mercury	{'MEA1':'Mercury	{'MGF':'Mercury	{'MIA':'Mercury I	{'MSA':'Mass Spe	{'MEA':'Mercury Electron Analyzer'}
Bepi	{'MAG':'Magnetc	{'ENA':'Energetic	{'MEA1':'Mercury	{'MEA1':'Mercury	{'MGF':'Mercury	{'MIA':'Mercury I	{'MSA':'Mass Spe	{'MEA':'Mercury Electron Analyzer'}
Mio	{'MAG':'Magnetc	{'ENA':'Energetic	{'MEA1':'Mercury	{'MEA1':'Mercury	{'MGF':'Mercury	{'MIA':'Mercury I	{'MSA':'Mass Spe	{'MEA':'Mercury Electron Analyzer'}
MPO	{'MAG':'Magnetc	{'ENA':'Energetic	{'MEA1':'Mercury	{'MEA1':'Mercury	{'MGF':'Mercury	{'MIA':'Mercury I	{'MSA':'Mass Spe	{'MEA':'Mercury Electron Analyzer'}
CALLISTO	spectrometer							
CANOPUS	ASI	BARS	MARI	MPA				
CHAMP	DIDM	{'FGM':['Flux Gat	OVM					
CNOFS	CINDI	CINDI	PLP	VEFI				

[Fig.12 – Excel spreadsheet extract for instruments]

- Regions spatiales

Asteroid						
Comet						
Earth	{'Magnetosheath':'Magne	{'Magnetosphere':{'Magn	{'Moon':'Moon'}	{'NearSurface':{'Atmosph	{'Surface':'Surface'}	
Interstellar						
Jupiter	{'Callisto':'Callisto'}	{'Europa':'Europa'}	{'Ganymede':'Ganymede	{'Io':'Io'}	{'Magnetosphere':{'Magnetotail':'Magnetotail','Main':'	
Mars	{'Deimos':'Deimos'}	{'Magnetosphere':{'Magn	{'Phobos':'Phobos'}			
Mercury	{'Magnetosphere':{'Magnetotail':'Magnetotail','Main'		{'Main','Plasmasphere':'Plasmasphere','Polar':'Polar','RadiationBelt':'RadiationBelt','RingCurrent':'RingCurrent'			
Neptune	{'Magnetosphere':{'Magn	{'Proteus':'Proteus'}	{'Triton':'Triton'}			
Saturn	{'Dione':'Dione'}	{'Enceladus':'Enceladus'}	{'Iapetus':'Iapetus'}	{'Magnetosphere':{'Magn	{'Mimas':'Mimas'}	{'Rhea':'Rhea'}
Sun	{'Chromosphere':'Chromo	{'Corona':'Corona'}	{'Interior':'Interior'}	{'Photosphere':'Photosph	{'TransitionRegion':'TransitionRegion'}	
Uranus	{'Ariel':'Ariel'}	{'Magnetosphere':{'Magn	{'Miranda':'Miranda'}	{'Oberon':'Oberon'}	{'Puck':'Puck'}	{'Titania':'Titania'}
Venus	{'Magnetosphere':{'Magnetotail':'Magnetotail','Main'		{'Main','Plasmasphere':'Plasmasphere','Polar':'Polar','RadiationBelt':'RadiationBelt','RingCurrent':'RingCurrent'			
Pluto						
Heliosphere	{'Heliosheath':'Heliosheat	{'Inner':'Inner'}	{'NearEarth':'NearEarth'}	{'Outer':'Outer'}	{'Remote1AU':'Remote1AU'}	

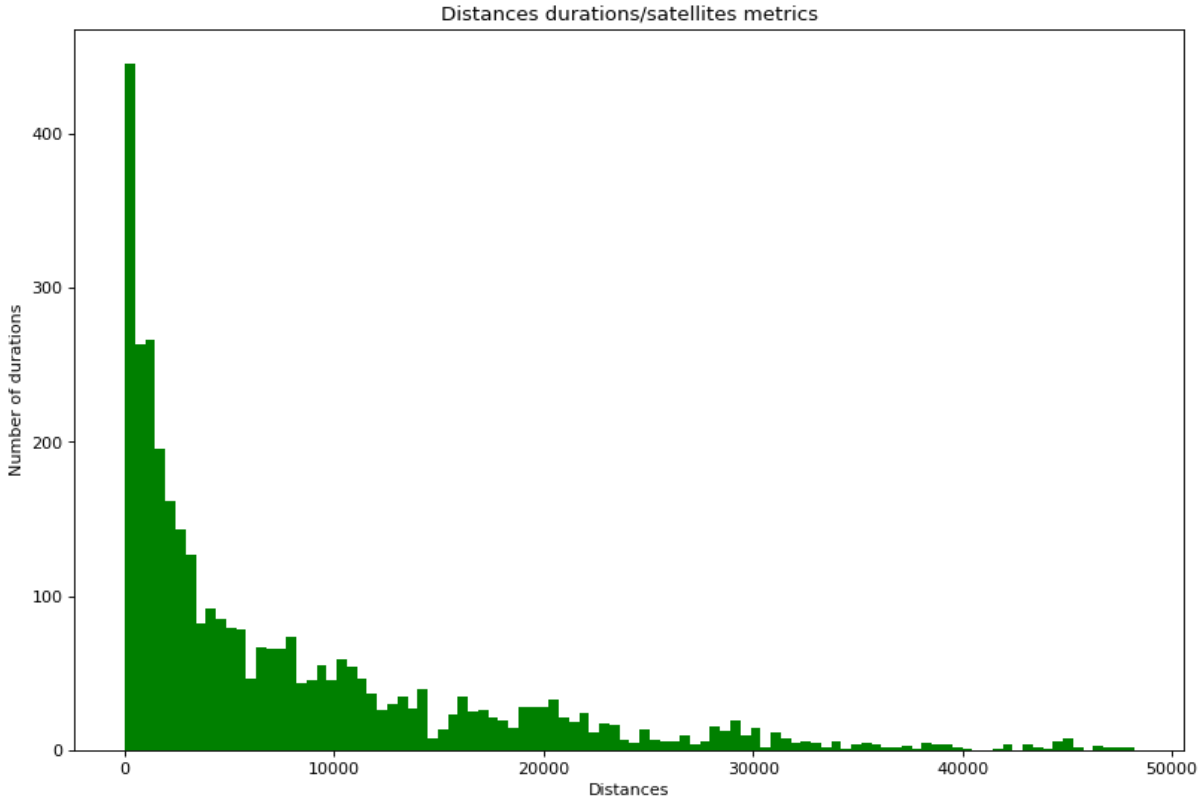
[Fig.13 – Excel spreadsheet extract of spatial regions]

In order to carry out entity recognition, the program scans each cell of the Excel sheet and uses regexes to find a match in the text resulting from occlusion (regexes are much faster and more efficient than a simple comparison search, which would require scanning all the words in the occluded text, for each entity searched). When an entity is found, it stores its position, i.e. the number of letters from the beginning of the file to the first letter of the entity found, and the number of letters from the beginning of the file to the last letter of the entity. Once all the entity cells in the Excel file have been browsed, the program can begin to link them together.

The idea behind the method chosen to link entities together is that when a human writes a sentence to say that something or someone has measured

something at such-and-such a place on such-and-such a date, the subjects (the entities) are never or rarely quoted very far from each other.

In order to verify this theory, we carried out a number of measurements ourselves to assess its consistency. These measures, reported below, express the distribution of distances between each type of entity.



[Fig.14 – Histogram of interval/satellite distance distribution]

In order to verify this theory, we carried out a number of measurements ourselves to assess its consistency. These measurements, reported below, express the distribution of distances between each type of entity.

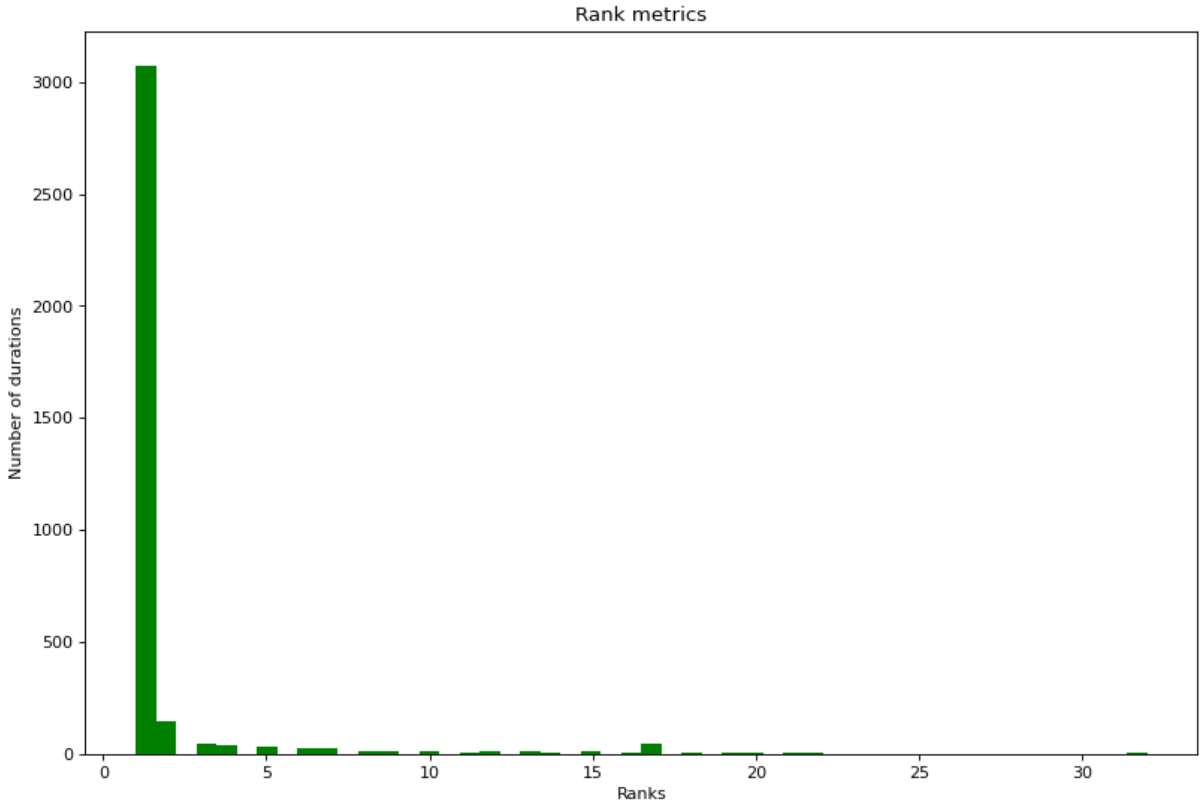
In order to ensure that a date is correctly linked to a mission, we have added a new sheet to our Excel file, including mission start and end dates (when available) (see illustration below), for each mission. If the closest time interval linked to a satellite is not within the mission's validity period, the program will link the mission with the next closest time interval, the third closest, and so on. As long as this is not valid.

BepiColombo	2018-10-19	
CARISMA		
CHAMP	2000-07-15	2010-09-19
CLIMAX		
CNOFS		
CORONAS-F	2001-07-31	
CRRES		
Cassini	1997-10-15	2017-09-15
Cluster-Rumba	2000-12-01	
Cluster-Salsa	2000-12-01	
Cluster-Samba	2000-12-01	
Cluster-Tango	2000-12-01	
Cluster	2000-12-01	

[Fig.15 – Excel spreadsheet extract of mission existence periods]

However, the question arises as to whether this approach might lead to less consistent results. Indeed, if the program finds a date within the mission period, but this is the tenth closest, there is little chance that this date is correct.

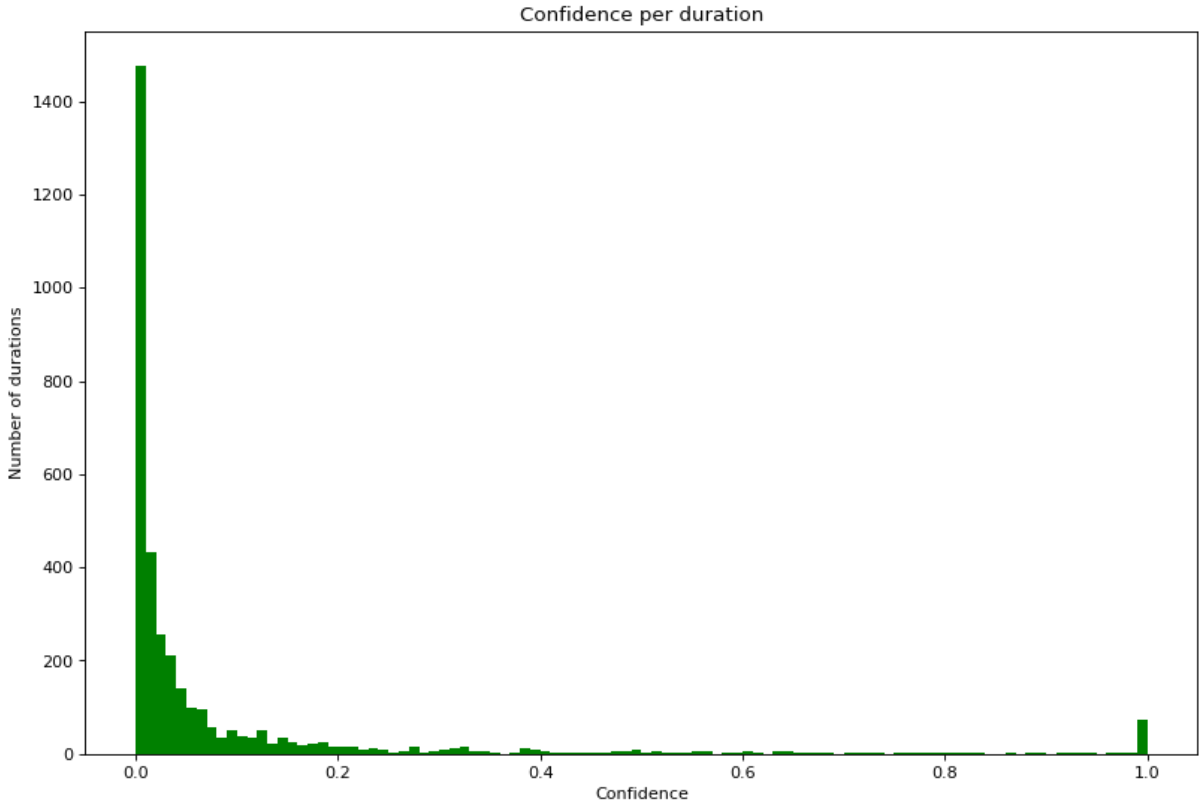
New measurements were then taken to check once again that this approach was consistent. By measuring the distribution of selected ranks for each time interval, we arrived at the following histogram:



[Fig.16 – Histogram of time interval rank distribution]

This graph clearly shows that the overwhelming majority of ranks are low, which means that it is very rare to recover a date quoted very far from the satellite detected by the reconnaissance program.

These two analyses combined have led us to develop a confidence index to measure, for each entity 1/entity 2 association, the confidence with which we can expect or not that this association is credible.



[Fig.17 – Histogram of entity association confidence index distribution]

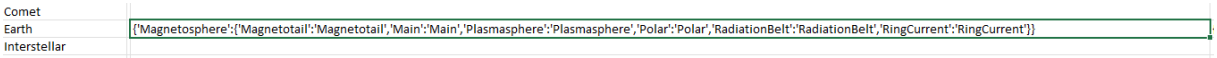
The lower the confidence index, the more credible the entity association. The results obtained from this analysis have enabled us to definitively validate the closest-quoted entity association approach.

However, there are still errors in the results. Indeed, as mentioned earlier, at this stage the program picks up ambiguities that shouldn't be there. As in the previous example, the program frequently confuses "wind" with "solar wind ion analyzer", for example. To deal with this case, we performed a check on the entities found. For each entity, since we have its position in the text, the program checks that an entity is not contained within a larger one, and if it is, it is simply removed from the list of entities found. Other ambiguities such as "The", confused with THEMIS-E sometimes abbreviated to "The", were noted. Such cases were handled by ignoring "The" if it occurs at the beginning of a paragraph or sentence.

As far as the association between satellite and instruments is concerned, we came up against a certain difficulty in being able to correctly link one to the other. The program does check whether an instrument actually belongs to the satellite with which it wants to associate it, but many satellites are equipped with the same instruments. So if in an article we have a mission 1 and 2 both equipped with the instrument, for example, "FPI", the program is not able to know whether in the context of the article we were talking about the "FPI" of mission 1 or 2 or both.

This problem led us to simply add all the instruments encountered for each satellite, and then delete all those that didn't belong to them respectively. This choice was based on the fact that the main and most important thing to do in terms of recognition and association for the success of the project was the satellites and their time intervals; possible errors on instruments and spatial regions were not a critical problem.

Finally, the recognition of spatial regions was also carried out by associating those quoted closest to a satellite. To do this, we first need to be aware that the formatting of spatial regions obeys SPASE rules. For example, if we're talking about the Earth's magnetosphere, it's written as earth.magnetosphere, whereas if we're talking about the plasmasphere, it's written as earth.magnetosphere.plasmasphere, the latter being a component of the magnetosphere. So, in order to establish a link between regions and satellites, the program searches for a satellite, then looks for the nearest low-level region name (which is not a planet), and finally for the planet name closest to the low-level name. Finally, in order to properly format the region found in SPASE format, we have set up a data structure in our Excel sheet that allows us to do this easily (see illustration below).



[Fig.18 – Illustration of spatial region data structure]

To explain this process without going into too much technical detail, let's imagine, as in a computer, that we have a main folder with subfolders named like the planets.

earth" is the name of a folder; "earth" contains the folder "magnetosphere", which in turn contains the folder "plasmasphere". which in turn contains the "plasmasphere" folder. So, if "plasmasphere" is found, with "earth" as the closest planet name, the program reconstructs the entire name SPASE in the same way as a computer reconstructs the path to a folder.

DOI

The final step in the textual processing of the article is to retrieve its DOI.

A first method devised to retrieve the DOI of an article was to identify it by regex. However, this method did not allow differentiation between the DOI of the article itself, and those of other articles cited in references or anecdotally. A solution to this problem could have been to retrieve only the DOI written at the beginning of the article, but as these publications were sourced by different publishers, with different page layouts, there was no guarantee that the DOI of the article being processed was written at the beginning of the publication. For this reason, this method was not adopted in the end.

A second method consisted in taking advantage of the structuring of XML formats to retrieve the DOI from the <idno type="DOI"> tag, itself contained in the article information tag <fileDesc>. To do this, we reused the aforementioned GROBID python package to generate a structured XML version of each publication using AI. After checking the accuracy of the results provided by this method, it became clear that the results were quite satisfactory. However, being a bit of a perfectionist, I thought it would be useful to look for a way of supplementing this approach to improve the results even further. We did this by retrieving the title of the article contained in the <title level="a" type="main"> tag, itself contained in the <fileDesc> tag. We then used the APIs of the NASA ADS website to request the DOI corresponding to the title of the publication, in the event that either the DOI was not present in the XML file, or was present, but following a check by doing the reverse, i.e. providing the DOI to the NASA ADS site, and requesting the corresponding title, this title did not correspond to that of the article. Finalement, l'utilisation de cette double méthode nous a permises sur nos 54 publications, de retrouver avec succès 52 DOI, soit une pertinence de 96%.

Writing the final result

Finally, the last stage of our project consists of writing our results in a text file, readable and processable by AMDA.

To do this, we must follow the conventions of the HPEvent format, which is structured as follows:

```
# Name: 10105100046361202140998_bibheliotech_V1;
# Creation Date: 2022-07-20T11:38:23.489308;
# Description: Catalogue of events resulting from the HelioNER code (Dablanc & Génot, "https://github.com/ADablanc/BibHelioTech.git") on the paper "https://doi.org/10.
first columns are the start/stop times of the event. the third column is the DOI of the paper, the fourth column is the mission that observed the event with the list c
the fifth column. The sixth column is the most probable region of space where the observation took place (SPASE ObservedRegions term);
# Parameter 1: id:column1; name:DOI; size:1; type:char;
# Parameter 2: id:column2; name:SATS; size:1; type:char;
# Parameter 3: id:column3; name:INSTS; size:1; type:char;
# Parameter 4: id:column4; name:REGS; size:1; type:char;
# Parameter 5: id:column5; name:D; size:1; type:int;
# Parameter 6: id:column6; name:R; size:1; type:int;
# Parameter 7: id:column7; name:SO; size:1; type:int;
# Parameter 8: id:column8; name:occur_sat; size:1; type:int;
# Parameter 9: id:column9; name:nb_durations; size:1; type:int;
# Parameter 10: id:column10; name:conf; size:1; type:float;
2020-07-01T00:00:00.000 2020-11-28T23:59:59.000 https://doi.org/10.1051/0004-6361/202140998 "ARTEMIS-P2" "ESA" "Earth.Magnetosphere" 1462 1 1 66 45 0.1518724354646029
2020-07-01T00:00:00.000 2020-11-28T23:59:59.000 https://doi.org/10.1051/0004-6361/202140998 "PSP" "LFR,FIELDS,RFS" "Sun" 6957 1 21 66 45 0.034413931780575946
2020-07-01T00:00:00.000 2020-11-28T23:59:59.000 https://doi.org/10.1051/0004-6361/202140998 "PSP" "LFR,FIELDS,RFS" "Sun" 484 1 21 66 45 0.00239418470343521
2020-07-01T00:00:00.000 2020-11-28T23:59:59.000 https://doi.org/10.1051/0004-6361/202140998 "STEREO-A" "" "Heliosphere.Remote1AU" 326 1 9 66 45 0.00376276135898013O7
2020-07-01T00:00:00.000 2020-11-28T23:59:59.000 https://doi.org/10.1051/0004-6361/202140998 "STEREO" "" "Heliosphere.Remote1AU" 6937 1 6 66 45 0.12010249484928756
```

[Fig.19 – Example of an extract from a final file in HPEvent format]

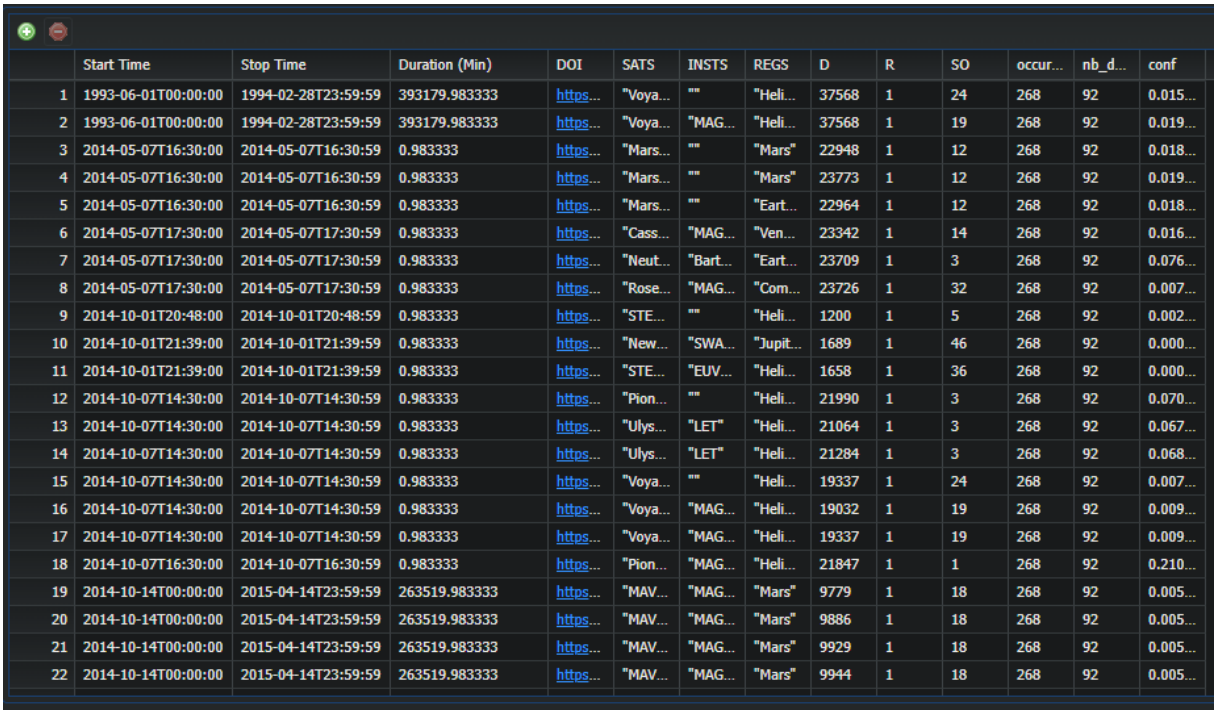
To do this, we must follow the conventions of the HPEvent format, which is structured as follows :

- A title
- A creation date
- A description
- One line for each parameter, these being the number of columns. Each parameter is associated with a column number, a name and a type.

Then, like an Excel spreadsheet (in CSV format), the rows are separated by line breaks, and the columns are separated by spaces.

Results obtained and their evaluation in relation to the initial need

At the end of this project, feeding AMDA with the HPEvent files we had generated gave the following result:

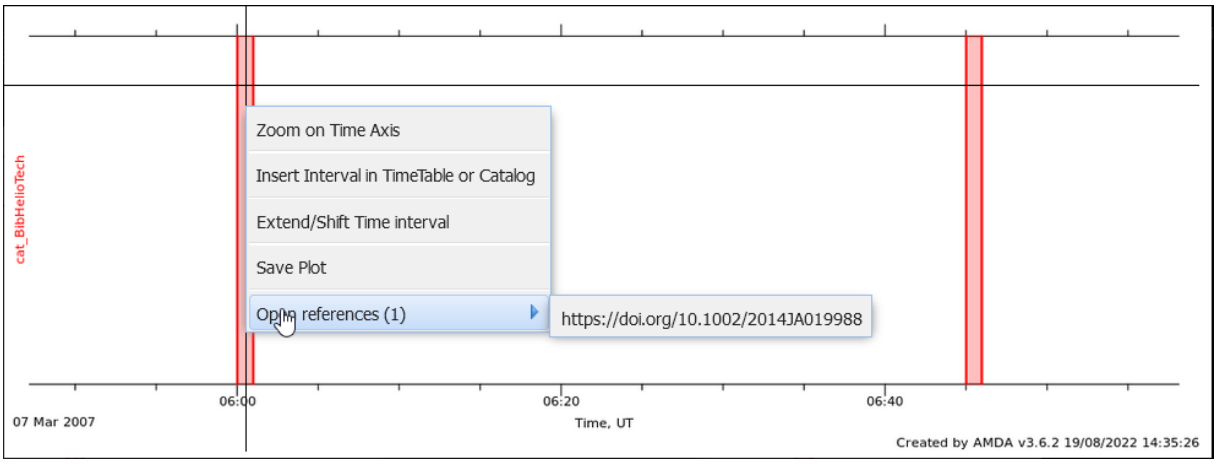


	Start Time	Stop Time	Duration (Min)	DOI	SATS	INSTS	REGS	D	R	SO	occur...	nb_d...	conf
1	1993-06-01T00:00:00	1994-02-28T23:59:59	393179.983333	https...	"Voya...	""	"Heli...	37568	1	24	268	92	0.015...
2	1993-06-01T00:00:00	1994-02-28T23:59:59	393179.983333	https...	"Voya...	"MAG...	"Heli...	37568	1	19	268	92	0.019...
3	2014-05-07T16:30:00	2014-05-07T16:30:59	0.983333	https...	"Mars...	""	"Mars"	22948	1	12	268	92	0.018...
4	2014-05-07T16:30:00	2014-05-07T16:30:59	0.983333	https...	"Mars...	""	"Mars"	23773	1	12	268	92	0.019...
5	2014-05-07T16:30:00	2014-05-07T16:30:59	0.983333	https...	"Mars...	""	"Eart...	22964	1	12	268	92	0.018...
6	2014-05-07T17:30:00	2014-05-07T17:30:59	0.983333	https...	"Cass...	"MAG...	"Ven...	23342	1	14	268	92	0.016...
7	2014-05-07T17:30:00	2014-05-07T17:30:59	0.983333	https...	"Neut...	"Bart...	"Eart...	23709	1	3	268	92	0.076...
8	2014-05-07T17:30:00	2014-05-07T17:30:59	0.983333	https...	"Rose...	"MAG...	"Com...	23726	1	32	268	92	0.007...
9	2014-10-01T20:48:00	2014-10-01T20:48:59	0.983333	https...	"STE...	""	"Heli...	1200	1	5	268	92	0.002...
10	2014-10-01T21:39:00	2014-10-01T21:39:59	0.983333	https...	"New...	"SWA...	"Jupit...	1689	1	46	268	92	0.000...
11	2014-10-01T21:39:00	2014-10-01T21:39:59	0.983333	https...	"STE...	"EUV...	"Heli...	1658	1	36	268	92	0.000...
12	2014-10-07T14:30:00	2014-10-07T14:30:59	0.983333	https...	"Pion...	""	"Heli...	21990	1	3	268	92	0.070...
13	2014-10-07T14:30:00	2014-10-07T14:30:59	0.983333	https...	"Ulys...	"LET"	"Heli...	21064	1	3	268	92	0.067...
14	2014-10-07T14:30:00	2014-10-07T14:30:59	0.983333	https...	"Ulys...	"LET"	"Heli...	21284	1	3	268	92	0.068...
15	2014-10-07T14:30:00	2014-10-07T14:30:59	0.983333	https...	"Voya...	""	"Heli...	19337	1	24	268	92	0.007...
16	2014-10-07T14:30:00	2014-10-07T14:30:59	0.983333	https...	"Voya...	"MAG...	"Heli...	19032	1	19	268	92	0.009...
17	2014-10-07T14:30:00	2014-10-07T14:30:59	0.983333	https...	"Voya...	"MAG...	"Heli...	19337	1	19	268	92	0.009...
18	2014-10-07T16:30:00	2014-10-07T16:30:59	0.983333	https...	"Pion...	"MAG...	"Heli...	21847	1	1	268	92	0.210...
19	2014-10-14T00:00:00	2015-04-14T23:59:59	263519.983333	https...	"MAV...	"MAG...	"Mars"	9779	1	18	268	92	0.005...
20	2014-10-14T00:00:00	2015-04-14T23:59:59	263519.983333	https...	"MAV...	"MAG...	"Mars"	9886	1	18	268	92	0.005...
21	2014-10-14T00:00:00	2015-04-14T23:59:59	263519.983333	https...	"MAV...	"MAG...	"Mars"	9929	1	18	268	92	0.005...
22	2014-10-14T00:00:00	2015-04-14T23:59:59	263519.983333	https...	"MAV...	"MAG...	"Mars"	9944	1	18	268	92	0.005...

[Fig.20 – Example of a catalog extract loaded into AMDA, from one of our final files]

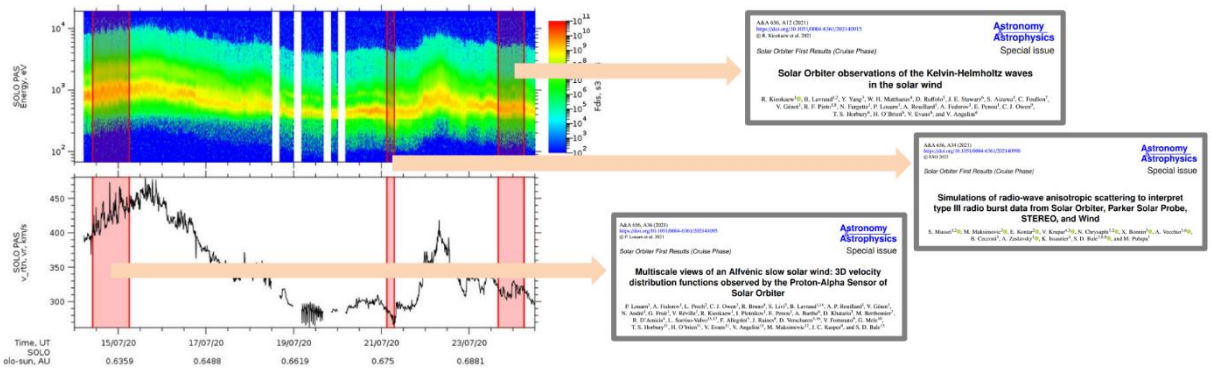
All the requirements necessary for the success of the project are present. Dates are sorted in ascending order. The durations of each interval have been calculated and are present. Satellites are present, with associated instruments, when cited in the original text. Each satellite is associated with a spatial region in SPASE format. And each association, or line, has a confidence index enabling the user to judge the credibility of the data he can observe. DOI links are present and clickable after the addition of this functionality by the IRAP AMDA team.

Visually, after plotting data in AMDA, the visualization interface looks like this:



[Fig.21 – Illustration of the visualization interface created by the AMDA team]

This is then superimposed on measurement data to obtain the final result:



[Fig.22 – Final customer requirement]

Reviews

Professional review

The absence of a centralized data repository for the various missions and studies carried out by researchers makes it difficult for them to detect physical measurements or phenomena already observed in the past. As a result, it remains complicated to realize that a phenomenon may, for example, occur cyclically every X amount of time, and even more difficult to obtain details concerning these phenomena, requiring us to reread a publication that must undoubtedly exist, but whose title or DOI we don't know. Thanks to our work, linking the work of heliophysics researchers around the world is made much more efficient. Of course, the results of this work are not 100% perfect. In the future, it would be wise to be able to create and train an AI capable of carrying out this work in a much more relevant way. Moreover, heliophysics is not the only field concerned by these problems of linking different publications. All other fields, whether computer science, oceanography or others, are also concerned. This work could be the beginning of a larger project, useful to all fields of research, worldwide.

Personal review

Having been fascinated by astronomy and astrophysics since childhood, I dream of one day working in an institute like IRAP. This internship has been a real treat for me. I'm extremely grateful to have had the chance and opportunity to discuss this field with IRAP researchers and other trainees. I was able to attend seminars, thesis defenses, BepiColombo's second flyby around Mercury, visit the clean rooms, tour the Titan room, IRAP's supercomputer, and even present my internship topic at the intern seminar organized by IRAP. But also to learn where to look for fixed-term contracts at the CNRS, or permanent contracts via the external competitive examinations, which I'll be sure to try next year. The IRAP staff were warm and welcoming, and I personally had a great time with them.

I also had a great time at work. I loved learning how to do text occlusion and structuring with XML; learning how to train an AI, even if we didn't actually use it; learning NER methods and temporal value recognition; and using API methods, which I'd already seen in class. This internship also enabled me to take my regex knowledge and know-how much further.

Conclusion

In short, the BibHelioTech program means :

- OCRising and xmlizing publications:
  - On the one hand, the program converts each page of an article into images, and on the other, recognizes the shapes of the letters to rewrite them in a text file.
  - On the other hand, it reads the raw article and structures the text in an XML file..
- Entity recognition:
  - The program reads the previously created text file and first recognizes the time values, then filters and transforms them to arrange them, finally formatting them in ISO 8601.
  - It then searches for the various satellites, instruments, etc., using the Excel file we've supplied for this purpose. Using the Excel file we've created for this purpose.
- Association of entities:
  - The program links the time intervals to the nearest satellites quoted in the text, so that they also fall within the mission validity period of said satellite.Puis, il relie les instruments et satellites, à condition que les satellites soient équipés des instruments en question.
  - Finally, it associates a satellite with a space region, the closest one cited, if this is one of the possible regions for this mission..
- DOI search:
  - The program retrieves the DOI from the publication's XML file. If the DOI does not correspond to the title of the publication, or is missing from the XML file, it queries the NASA ADS APIs to find the correct DOI based on its title. It then adds it to the final results.
- Writing HPEvent:
  - Finally, the final results are written in HPEvent format, readable by AMDA, in files ready to be loaded onto AMDA by a user..

Bibliography et sitography

- PyPI - pytesseract · PyPI, [en ligne] _ https://pypi.org/project/pytesseract/

- Tesseract-Ocr - Tesseract User Manual | tessdoc, [en ligne] _ https://tesseract-ocr.github.io/tessdoc/
- Github - kermitt2/grobid: A machine learning software for extracting information from scholarly documents, [en ligne] _ https://github.com/kermitt2/grobid
- nlp.stanford - The Stanford Natural Language Processing Group, [en ligne] _ https://nlp.stanford.edu/software/sutime.shtml
- PyPI - sutime · PyPI, [en ligne] _ https://pypi.org/project/sutime/1.0.0rc2/
- nssdc.gsfc.nasa - Welcome to the NSSDCA, [en ligne] _ https://nssdc.gsfc.nasa.gov/
- ui.adsabs.harvard – NASA/ADS, [en ligne] _ https://ui.adsabs.harvard.edu/
- ISTEX - Le socle de la bibliothèque scientifique numérique nationale - Istex, [en ligne] _ https://www.istex.fr/
- Hpde - HPDE.io, [en ligne] _ https://hpde.io/
- AMDA - Welcome on Amda, [en ligne] _ http://amda.irap.omp.eu/
- Zenodo - ADablanc/BibHelioTech: Version 3.7.11 | Zenodo, [en ligne] _ https://zenodo.org/record/6867940#.YwtRnHZByUk
- Github - ADablanc/BibHelioTech, [en ligne] _ https://github.com/ADablanc/BibHelioTech