INDIAN INSTITUTE OF TECHNOLOGY
BOMBAY

RnD PROJECT REPORT

Automatic extraction of coordinates from ATels

Submitted To:

Prof. Amit Sethi
Electrical Engineering Department
IIT Bombay

Submitted By:

Naman Jaswani
193070052
M.Tech EE - IIT Bombay

Table of Contents

Introduction

This project aims to automatically extract the location of objects referenced in astronomy circulars called ATels.

***"The Astronomer's Telegram** (**ATel**) is an internet based short notice publication service for quickly disseminating information on new astronomical observations. Telegrams as available instantly on the service's website and distributed to subscribers via email digest within 24 hours.–*

The Astronomer's Telegram was launched on 17 December 1997 by Robert E. Rutledge with the goal of rapidly (<1 s) sharing information of interest to astronomers. Telegrams are sent out daily by email, but especially time sensitive events can be transmitted instantly."

In observational astronomy, one cannot wait for a proper peer review process before publishing a discovery like that in other sciences, be it experimental physics, or medical sciences, or mathematics, etc. This is because discovery events in observational astronomy are many times very short in time scale and by the time one waits for the peer review, the event has already faded away.

Also, the field is extremely collaborative and there is often a need to observe an event in multiple wavelengths, from radio to infrared to visible light all the way to ultraviolet and x-ray so that one can get maximum possible data to understand the event. Therefore, it is very important that the discovery is published in real-time so that all the other collaborators are able to aid the observation of the event in real-time.

The above constraints results in the lack of a proper peer review process, there are many cases when the astronomers do not read the recent circulars and report the same discovery again.
Also, many times astronomers observe something in a particular location in the sky and they want to access all circulars which referred to this particular region so that it can aid their research.

The variety of events taking place does not allow the community to use a fixed rigid format for reporting their discoveries. Also, there are a large number of different types of formats in which astronomers report the location of their discovery. As a result, the text in such circulars cannot be directly parsed to extract the location information.

To address such problems there is a need to develop an interface where the researchers can enter the coordinates of the location they are interested in and get all the circulars which refer to this particular location with an error. In this project, we developed a tool which uses regular expressions and natural language processing methods to extract all the coordinates from a given ATel.

Problem Definition

The project aims at extracting the location of the observed objects from astronomy circulars in an automated way. This would be used to create an interface to query all the circulars which referred to a particular location.
The location of any object in the sky is given by a pair of coordinates called right ascension (R.A) and declination (Dec.). Just as every point on the surface of the earth can be specified using its latitude and longitude, every point in the sky is specified by its right ascension and declination. The right ascension corresponds to the longitude and can be specified in either degrees (0 to 360) or hours (0 to 24). The declination corresponds to the latitude and is specified in degrees (-90 to 90). There are many intricacies about the astronomical coordinates but they are not relevant from the point of this problem statement.
The input is the text content of the ATel, from which we want to extract coordinates information, and the output is a set of coordinate pairs from that ATel.

Examples

1. http://www.astronomerstelegram.org/?read=13354

Inference:

The first paragraph gives the coordinates ie " R.A.: 60.43819 deg, Dec.: 21.17461 deg "

The output should be {('60.43819', '21.17461')}

2. http://www.astronomerstelegram.org/?read=13361

Inference:

There is no explicit mention of coordinates however the coordinates can be extracted from the object name ie "J060000.76-310027.83" in the first line of the first paragraph. From this, it can be inferred that the object has Ra = 06h 00min 00.76sec and Dec = - (31deg 00 min 27.83sec). However, in the first example also we had such object names but since explicit coordinate names are more accurate than those inferred from object names, we should report that.

The output should be {'060000.76-310027.83'}

3. http://www.astronomerstelegram.org/?read=13351

Inference:

This has no information about coordinates explicitly or implicitly.

The output should be {} i.e. an empty set

4. http://www.astronomerstelegram.org/?read=13330

Inference:

This has many coordinates given in the tabular format and all need to be extracted.

5. http://www.astronomerstelegram.org/?read=13347

Inference:

This also has no coordinates but illustrates that if we use regex there will be false matches like "2019-12-06 13:10:19 to 2019-12-07 09:28:36 (UTC)" from the first sentence of the first para.

The output should be {} i.e. an empty set

Approaches:

Deep learning

The very first approach to attack the problem was using Named Entity Extraction (NER) by deep learning methods. This is because at first glance it looked like a simple NER problem with coordinates as named entities, which can then be easily recognized by NER using the context of the input sentence, and hence be easily extracted. Myself and Adeem jointly labelled a dataset using the online annotator, LightTag, on which we had planned to implement transfer learning of language models like BERT.

Although we divided the work, the process of labelling the dataset by going through the ATels one at a time was extremely slow and we couldn't build a large enough dataset. Also going deep into its implementation, we figured out that we could not find a pretrained vocabulary for the scientific text of ATels. Another problem which was faced using deep learning was that we could not find a suitable tokenization to train the vocabulary from scratch. Also, there was the realisation that according to the complexity of our task training a new vocabulary from scratch was unnecessary.

I built a basic BiLSTM to test our hypothesis, but the model (NER) didn't train at all (as expected) due to small amount of dataset that we were using.

Traditional Machine Learning

While I was trying his best to make deep learning work, Adeem decided to look for simpler methods to tackle the problem, even though we kept on discussing our approaches and ideas. We tried to see how far can we go using regular expression methods because almost always the coordinate information comes in the following formats:

Formats

Category 1

Ehhmmss.ss+ddmmss.s	J180420.99-293108.9

Category 2

RA (hours)	Dec (deg)	Example
hh:mm:ss.ss	dd:mm:ss.ss	18:04:20.99 -29:31:08.9
hh mm ss.ss	dd mm ss.ss	18 04 20.99 -29 31 08.9
hh:mm:ss.ss,	dd:mm:ss.ss	18:04:20.99,-29:31:08.9
00h 00m 00.0s	00d 00m 00.0s	18h 04m 20.99s -29d 31m 08.9s
00h00m00.0s	00d00m00.0s	18h04m20.99s -29d31m08.9s
00h00m00.0s	00°00'00.0"	18h04m20.99s -29°31'08.9"
hh.hhhhhhh	dd.ddddddd	18.072497 -29.519139

RA (deg)	Dec (deg)	Example
d dd:mm:ss.ss	dd:mm:ss.ss	d 271:05:14.85 -29:31:08.9
d dd mm ss.ss	dd mm ss.ss	d 271 05 14.85 -29 31 08.9
00d 00m 00.0s	00d 00m 00.0s	271d 05m 14.85s -29d 31m 08.9s
00d00m00.0s	00d00m00.0s	271d05m14.85s -29d31m08.9s
00°00'00.0"	00°00'00.0"	271°05'14.85" -29°31'08.9"

d dd.ddddddd	dd.ddddddd	d 271.087458 -29.519139

l (deg) b (deg) converts also to galactic coordinates

Problems after using only regex

We implemented regular expressions for all the formats in Category 2 except the decimal format.
We faced three problems while using only a regex-based approach:

1. Could not write a regular expression for cases where coordinates occur in decimal numbers without getting a huge number of false positives.

2. The format where there is a colon gave a significant number of false positives with date-time formats.

Example: 'INTEGRAL performed a target-of-opportunity observation of the colliding wind binary eta Carinae from 2019-12-06 13:10:19 to 2019-12-07 09:28:36 (UTC)'.
The colon format would give false positives here and would extract 13:10:19, 09:28:36

3. The object name format gave a moderate number of false positives with ranges.

Example: 'We obtained optical spectra covering the range 5000-9400 A with the 3.58m TNG telescope equipped with LRS at Observatorio del Roque de los Muchachos in La Palma (Spain) on Dec 14, 20:13 UT'.
5000-9400 would be extracted and it is a false positive.

Solution to overcome Regex problems

While going through the various examples of sentences having coordinates, We divided them into two categories.
Category 1) Sentences containing the object name format
Category 2) Sentences containing all other formats
Category 1 and 2 have different content with respect to each other but the content is similar in a single category. Refer appendix for examples of category 1 and 2.

Therefore, We planned to build two classifiers (Classifier A and Classifier B) to tackle the above problem.
Classifier A would classify a sentence as containing object name format match as a false positive or true positive.
If there is a match using the object name format then the sentence is passed to classifier A and if the label is 1, we append it to the output list.

Example:

1. We would like to thank the Swift team and other colleagues for providing continuous Swift monitoring observations of MAXI J1820+070 to the public.

This example is a true positive match and contains coordinate information in the '1820+070' substring.

2. All observations were performed on the ESO New Technology Telescope at La Silla on 2018 April 5 UT, using EFOSC2 and Grism 13 (3985-9315A, 18A resolution).

This gives a match '3985-9315' which is a false positive.

3. According to our photometry, 3C 279 shows a brightening of about 1 mag in NIR and 0.6-0.8 mag in optical with respect to the mean values observed by REM during 2005-2012 (see Sandrinelli, et al.2016, ApJ, 151, 54), in agreement with the increase of gamma-ray activity observed by Fermi-LAT.

This example gives a match '2005-2012' which is also a false positive.

Classifier B would classify a sentence as having coordinate information (in a format other than the object name format)

The role of the classifier is 2-fold

1. It would help in extracting decimal coordinates in cases where a sentence is labelled as 1 but has no matches with the regular expression function we had built.

2. Remove false positives with the time format as such sentences would be labeled as 0 by the classifier.

Examples:

1. The Large Area Telescope (LAT), one of two instruments on the Fermi Gamma-ray Space Telescope, has observed increasing gamma-ray emission from a source positionally consistent with the flat spectrum radio quasar object TXS 0358+210 (also known as MG2 J040146+2110, GB6 J0401+2110 and 4FGL J0401.7+2112, Fermi-LAT Coll. 2019arXiv190210045T) with radio coordinates (J2000) R.A.: 60.43819 deg, Dec.: 21.17461 deg (Beasley et al. 2002, ApJS 141, 13). TXS 0358+210 has a redshift z=0.834 (Sowards-Emmerd et al. 2005, ApJ 626, 95).

This should be classified as label 1 as it contains coordinate information. Since, it has no matches from the other formats of category 2(except decimal format), decimal number matching would be applied and the required coordinates would be extracted.

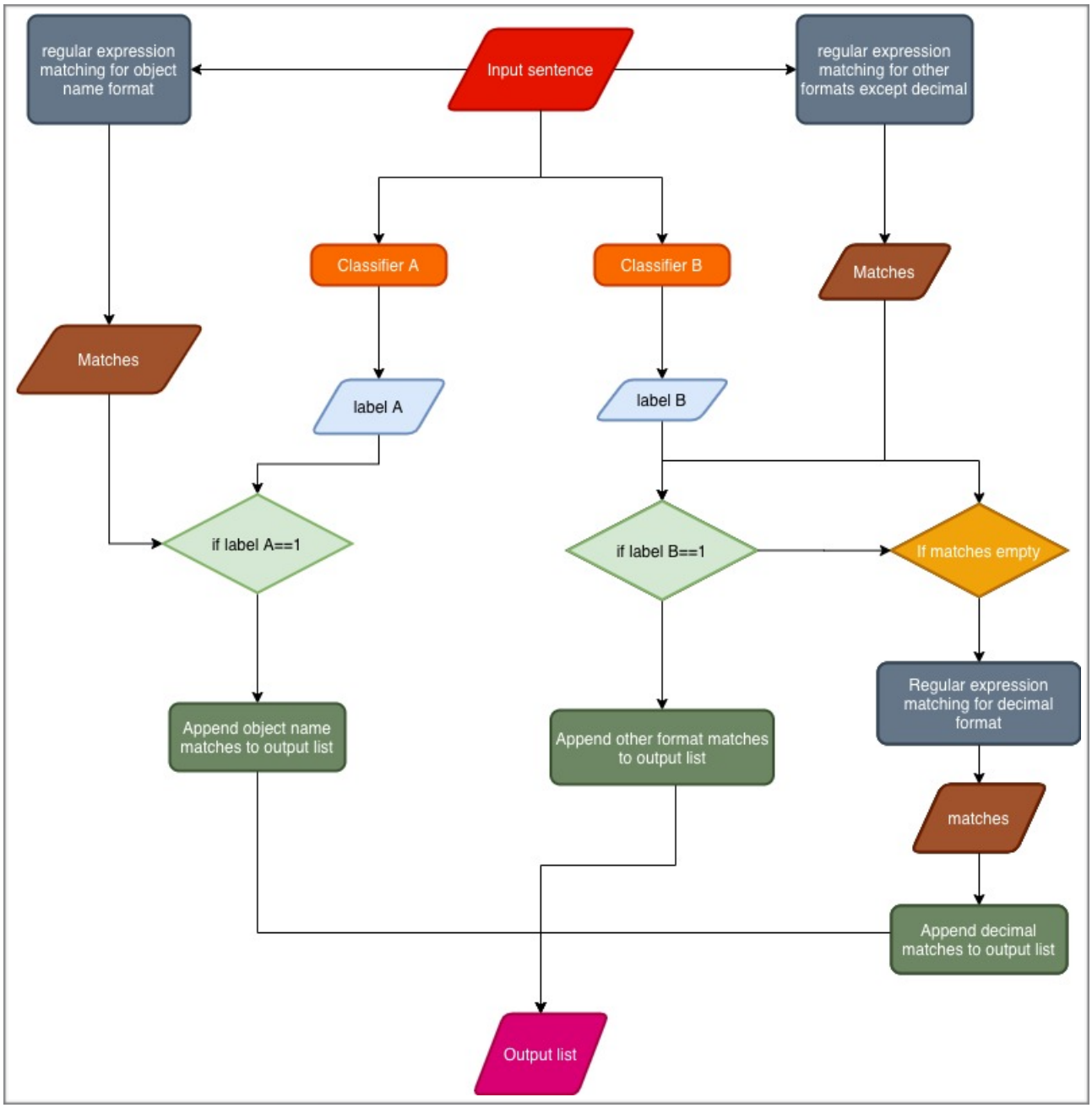
2. INTEGRAL performed a target-of-opportunity observation of the colliding wind binary eta Carinae from 2019-12-06 13:10:19 to 2019-12-07 09:28:36 (UTC).

This should be classified as label 0 as it does not contain coordinate information

If the sentence is classified as positive by classifier B and has matches with other formats of category 2 then those matches are appended to the output. If the sentence is classified positive but has not matches with other formats of category 2 then it must contain the decimal format and we apply decimal number matching (pair of decimal numbers) to extract the coordinates.

Note that We decided not use the classifier A in our current pipeline due to reasons related to limited amount of data for classifier A, which will be explained further in methods section.

Flowchart


regular expression matching for object name format
Input sentence
regular expression matching for other formats except decimal
Classifier A
Classifier B
Matches
Matches
label A
label B
if label A==1
if label B==1
If matches empty
Append object name matches to output list
Append other format matches to output list
Regular expression matching for decimal format
matches
Append decimal matches to output list
Output list

Methods

Regular expression and get_coordinates function

The get_coordinates function returns a list of all the RA, Dec pairs in a particular sentence which match the given list of regular expressions.
In this project there are three such functions:

1. For matching the object name format (Category 1)
2. For matching ra, dec pairs from the Category 2 except for the decimal format
3. For matching a pair of decimal numbers

For more details of the function please refer to the required_functions.py in the github repo.

Building and analysing Dataset for Object Name format (Classifier A)

We decided to use our function which extracted Object Name format using regular expression to interactively create the dataset. We jointly labelled the dataset by running a code to iterate through the ATels and whenever the function got a match we interactively labelled the match as either 1 if it is a true positive, 0 if it is a false positive. The code was run until we got around 600 examples and this constituted our dataset for Classifier A.

Understanding the dataset:

The dataset had 484 positive examples and 109 negative examples. Among the 109 negative examples 59 were almost identical and had the keyword ‘grism’.
Example: The observations were performed on the ESO New Technology Telescope at La Silla on 2018 April 8 UT, using EFOSC2 and grism 13 (3985-9315A, 18A resolution).
We decided to filter these examples, after which, the dataset had only 50 negative examples. This is a highly skewed dataset and training a model on this data gave very poor results resulting in many false negatives (the details of the model will be explained in the next section on Classifier B as we used the same model and pre-processing to train both Classifiers). From the perspective of our problem statement, avoiding false negatives is of utmost importance.
Without any classifier the accuracy was 92 percent with zeros false negatives and 8 percent false positives. So, We decided not to use Classifier A and output all matches from the regular expression and filtering for the keyword ‘grism’.

Building and analysing data set for Classifier B:

Trick used to make the dataset: One of the main objectives of building the classifier B is to be able to extract decimal formats. However, if I run a decimal matching regex to build the dataset there will be a lot of false positives in the dataset and hence an uncontrolled class imbalance. I used the fact that the content of all sentences in Category B is similar and build the dataset using all formats from Category B except the decimal format.
The function which extracted all formats except the decimal was used here. We jointly labelled the dataset by running a code to iterate through the ATels and whenever the function got a match, we interactively labelled the match as either 1 if it is a true positive, 0 if it is a false positive, and 2 if the match was part of a table instead of a sentence. As expected, almost all of the false positives were of the colon format which matched with the date-time format. Then we filtered the sentences with label 2.

Fixing class imbalance

At this point, there were 286 positive labels and 117 negative labels. To fix the class imbalance, I decided to explicitly add more negative examples. Since one of the purposes of this classifier is to extract coordinates in the decimal form, we need to make sure that the model does not classify sentences with other decimal numbers as positive because in that case, we would have had many false positives. Therefore, I used a function which matched decimal numbers (in a pair) and iterated through the ATels and interactively labelled the matches as 0 if it is a false positive, 1 if it is true positive (coordinates) and 2 if the match was from a table. As expected, after filtering deleting the sentences with 2, the 275 of the matches were 0 and only 27 were 1.
The final dataset was constructed by merging the initial dataset and the negatives from the decimal matching. This was a nice balanced dataset having 286 positives and 392 negatives.
The reason behind choosing decimal format for creating negative emamples is that we do not want our model to classify a sentence with a non-coordinate decimal number with label 1, while all other formats were all labeled accordingly.

Building model

Our machine learning pipeline consists of 3 components:

1. Countvectoriser (Bag of words representation)
2. Tf-idf Transform
3. Linear SVM Classifier

Bag of words representation

We decided to use a bag of words representation for the sentences. The bag of words representation consists of two steps during training: the first step is featurization . The featurization method has an argument called max_features. This step makes a vocabulary from our entire list of training sentences using the top most frequently occurring words (max_features).
The next step is fitting the featurizer to a list of sentences to vectorize them. For each sentence a vector of length max features would be created whose every entry would correspond to a word in the vocabulary of the featurizer. Each entry of this vector would be equal to the frequency of the word in the sentence.
Reason for Choice of representation.
Going through the positive examples we saw that all contain some keywords in the sentence. There were keywords which either point to the fact that this sentence contains coordinates or keywords like UTC, date, time which point to the fact that this is a false positive The important observation here was that to classify a sentence its content was important and not the specific arrangement of the words. Therefore, we used a simple bag of words model to vectorise our sentences.

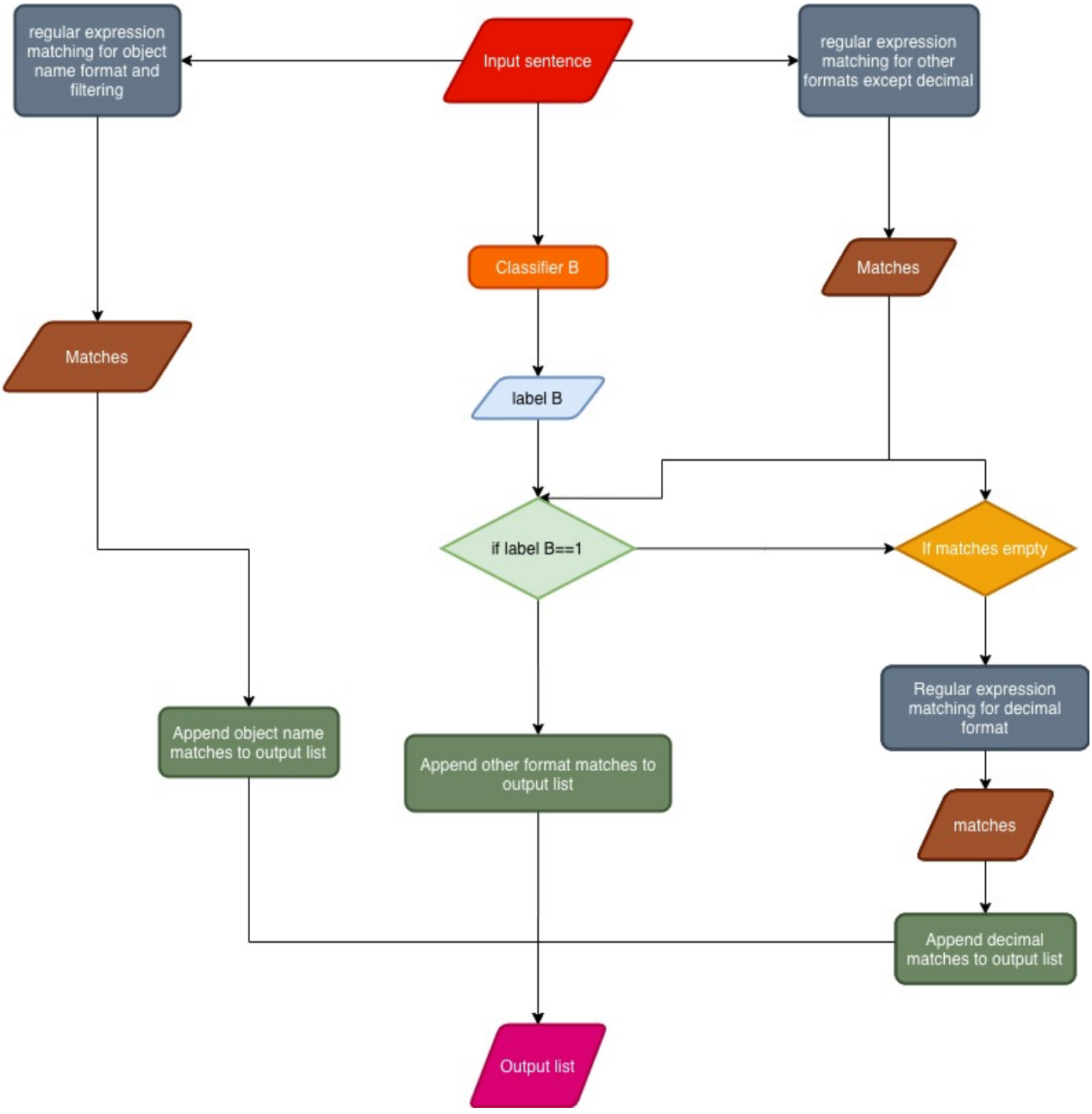
Tf-idf : The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

Linear Support Vector Machine is widely regarded as one of the best text classification algorithms.

Pre-processing:

Before passing the sentences to the Countvectorizer all characters except alphabets and '.', were removed, all characters were converted to lower case and all stop words were removed. There are more details of the pre-processing specific to our data
For more details refer to the
process_sent function in the required_functions.py file in the github repo.

Final Pipeline



Results

Metrics for Classifier B

Test results

Accuracy = 95.18%

	Precision	Recall	F1 Score	Support
0	**0.96**	**0.95**	**0.96**	**145**
1	**0.94**	**0.95**	**0.95**	**125**

TN =138	**FP =7**
FN =6	**TP =119**

Train results

Accuracy = 99.23%

	Precision	Recall	F1 Score	Support
0	**1.00**	**0.99**	**0.99**	**242**
1	**0.98**	**0.99**	**0.99**	**163**

TN =240	**FP =2**
FN =1	**TP =162**

Classifier Model is able to catch false positives:

Although the following sentences give match with regex but the model labels it as 0

1. ‘We obtained 8.6 ksec of NICER data between 2018 November 14 at 22:11:16 UTC and 2018 November 15 at 15:36:13 UTC.’
2. ‘The object was observed at 17.0-17.2m during 9 minutes from 22:15:26 to 22:24:30 UT on 2018-04-26, but was below the detection limit (18.8m) 45 minutes before the first positive observation and 43 minutes after the last one.’

Model is able to extract decimal formats:

Although the model was trained on formats of Category 2 other than decimal, it is able to identify sentences which contain coordinates in decimal format.

1. Based on the pointing direction of Astrosat at the time of the GW event (RA = 189.2, DEC = 62.3), the FRB was 157 degrees off axis.
2. At the instant of the FRB, AstroSat was pointing at (RA = 189.2, DEC = 62.3): about 157 degrees away from the nominal FRB direction.

Performance on examples directly from Problem statement:

1. http://www.astronomerstelegram.org/?read=13354

Output is {(' 60.43819', ' 21.17461'), '0358+210', '0401+2110', '0401.7+2112', '040146+2110'}

2. http://www.astronomerstelegram.org/?read=13361

Output is {'060000.76-310027.83', '2017-84089.'}

The second match is a false positive.

3. http://www.astronomerstelegram.org/?read=13351

Output is {'5000-9400'}

This is a false positive.

4. http://www.astronomerstelegram.org/?read=13330

We have not yet considered the table format.

5. http://www.astronomerstelegram.org/?read=13347

Output is {}

Future work:

1. By generating more data, the classifier A can be improved significantly
2. Extracting coordinates and/or other relevant information from tables.
3. There is another category of coordinates in which an event can be represented in, called Galactic Coordinates. This can be extracted using the same principles used in this project.
4. Important information from ATels, other that coordinates, like object name, error range, intensity can be extracted as per requirement.
5. Deep Learning models like (NER) can be used if larger amount of data is provided.

Acknowledgements

1. Adeem Jassani, my project partner for constant collaboration in discussing ideas and implementation.
2. Professor Amit Sethi to give me his guidance and opportunity to work on a project which can add real value to researchers.
3. Professor Varun Bhalerao and Prof Ashish Mahabal for the guidance and feedback.
4. Vihari Piratla, PhD student from CSE department for his guidance and time.

Appendix

Examples of category 1 and 2

Category 1: Object name format

1)The Nordic Optical Telescope (NOT) Unbiased Transient Survey (NUTS; ATel #8992) reports the spectroscopic classification of Gaia17dkc / SN2017jfo in host galaxy SDSS J013336.84+332552.0.

2)The Large Area Telescope (LAT), one of two instruments on the Fermi Gamma-ray Space Telescope, has observed gamma-ray emission from a source positionally consistent with the radio source PKS 2247-131, with coordinates RA=342.4983854 deg, Dec=-12.8546736 deg (J2000; Beasley et al.2002, ApJS, 141, 13), and no measured redshift.

3)Although the field is heavily polluted by single scattered photons from nearby bright LMXB GX 5-1 we clearly detected the source, with best-estimated position of (J2000) 269.23892, -25.10790 (error is 2.5", 90% confidence), which is 0.65" from catalog position of SWIFT J1756.9-2508.

4)Following the report of a new outburst of the accreting millisecond X-ray pulsar Swift J1756.9-2508 (ATel #11497), NICER performed pointed observations starting on 2018 April 3, collecting 9.4 ks of exposure over the ~30 hours between April 3 15:18 UTC and April 4 21:01 UTC.

5)We report the V-band photometric observation of MAXI J1820+070 (also named ASASSN-18ey, see ATELs #11399, #11400, #11403, #11404, #11406, #11418, #11420, #11421, #11423, #11424, #11425, #11426, #11432, #11437, #11439, #11440, #11445, #11451, #11478, #11481) with the Lijiang 2.4m telescope (+YFOSC) at Lijiang Gaomeigu Station of Yunnan observatories, in comparison with the Swift/XRT observation of MAXI J1820+070 on the same day.

Category 2: Other formats

1)The Large Area Telescope (LAT), one of two instruments on the Fermi Gamma-ray Space Telescope, has observed gamma-ray emission from a source positionally consistent with the radio source PKS 2247-131, with coordinates RA=342.4983854 deg, Dec=-12.8546736 deg (J2000; Beasley et al.2002, ApJS, 141, 13), and no measured redshift.

2)Although the field is heavily polluted by single scattered photons from nearby bright LMXB GX 5-1 we clearly detected the source, with best-estimated position of (J2000) 269.23892, -25.10790 (error is 2.5", 90% confidence), which is 0.65" from catalog position of SWIFT J1756.9-2508.

3)The slew position is RA: 05h36m29.4s, DEC: -67d59m40s with a 1-sigma error circle of 8 arcseconds, lying at 7 arcseconds from the XMM-Newton serendipitous catalogue source, 3XMM J053630.3-675935, which with a flux Fx~1.1E-14 ergs/s/cm2 (2012-03-24), is a factor ~200 fainter than the slew measurement.

4)The reported coordinates are consistent with those of a bright (I = 15.04 mag), red (V-I = 3.07 mag) star OGLE-LMC173.6.14223 (R.A. = 05:36:30.20, Decl.= -67:59:37.0, J2000.0).

5)The Large Area Telescope (LAT), one of two instruments on the Fermi Gamma-ray Space Telescope, has observed an intense gamma-ray flare from a source positionally consistent with the flat spectrum radio quasar 3C 279, also known as 3FGL J1256.1-0547 (Acero et al.2015, ApJS, 218, 23), with radio coordinates R.A.: 12h56m11.1665s, Dec: -05d47m21.523s (J2000.0; Johnston et al.1995, AJ, 110, 880).