Requirements from Unstructured Text

Jérémie Huppé
Polytechnique Montreal
Montreal, Canada
jeremie.huppe@polymtl.ca

KEYWORDS

Software requirement, unstructured text, text analysis, natural language processing (NLP), machine learning, unstructured data, conversation in text form.

ACM Reference format:
Jérémie Huppé. 2020. Requirements from Unstructured Text. In *Proceedings of ICSE '20: ,*
, Seoul, Korea (ICSE '20), 15 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The goal of this project is to find needs expressed in unstructured conversations and mapped them to software requirements.

2 DEFINITIONS

- **Forum**: A website or web page where users can post comments about a particular issue or topic and reply to other users' postings [5].
- **Need**: A thing that is wanted or required [5].
- **Subforum**: A section of an online forum dedicated to a specific topic, typically one relating to that of the main forum [5].
- **Thread**: A group of linked messages posted on an Internet forum that share a common subject or theme [5].
- **Post**: A piece of writing, image, or other item of content published online, typically on a blog or social media website or application [5].
- **Need keyword**: Word that can be find in a sentence that express a need.

3 DATA PROCESS

1. **Goal**: Identify if there are needs expressed in forums.
   - 1.1. **Procedure**: Read through different types of forums like TripAdvisor Forum [1], Diabetes.co.uk Forum [4] and Reddit Forum [8] and tried to find needs expressed in discussions.
   - 1.2. **Relevant Insights & Results**: Found that in certain type of forums there were needs expressed. Those needs were expressed differently depending on the type of forums and the context.
   - 1.3. **Conclusion**: There are needs expressed in certain types of forums.
2. **Goal**: Identify the different ways in which a need can be expressed.
   - 2.1. **Procedure**: Read through the Diabetes.co.uk Forum [4] and extracted sentences that expressed a need.
   - 2.2. **Relevant Insights & Results**: Found that needs can be communicated under two types of sentences: interrogative sentences and non-interrogative sentences. Also found that the extracted sentences that expressed a need had a set of keywords in common such as; *how*, *can't find*, *problem* and more.
   - 2.3. **Conclusion**: Needs can be communicated under interrogative and non-interrogative sentences and have a set of keywords in common.
3. **Goal**: Extract a list of keywords that are used to express needs.
   - 3.1. **Procedure**: Extracted manually the relevant keywords from the extracted sentences that expressed a need. The post source and the sentence source where each keywords were found is detailed in the dataset 4.4.
   - 3.2. **Relevant Insights & Results**:
     51 keywords were extracted.
     The extracted list of keywords will be referred to as *list of need keywords* see dataset 4.5.
     Found that some keywords were more frequent than other.
     The same keyword can be used in a sentence that express a need and also in a sentence that does not.
     Because only a sample of discussion has been analyzed the list of keywords is probably incomplete.
   - 3.3. **Conclusion**: A list of need keywords has been created see dataset 4.5. This list of keywords is probably incomplete (there is no official way of checking if this list is complete or not).
4. **Goal**: Expand the list of need keywords.
   - 4.1. **Procedure**: Using NLTK WordNet [3] all the synonyms of every initial keywords in the *list of need keywords* see dataset 4.5 have been found and added to it.
   - 4.2. **Relevant Insights & Results**: 196 synonyms of the initial keywords were found and added to the *list of need keywords* see dataset 4.5 which now contains 247 keywords see dataset 4.6.
     Some synonyms of the initial keywords does not seem to be related to sentences that express a need, words such as: *motor*, *attack*, *business*...
     On the other hand some synonyms of the initial keywords seems to be related to sentences that express a need: *search*, *upset*, *issue*...
     This new list of need keywords might still be incomplete.
   - 4.3. **Conclusion**: The list of need keywords has been extended see dataset 4.6. This list of keywords might have irrelevant keywords and might be incomplete.
5. **Goal**: Extract large quantity of data from Diabetes.co.uk Forum [4] for further analysis.
   - 5.1. **Procedure**: Scraped all the data of Diabetes.co.uk Forum [4] starting from subforums up to post messages between the 19-06-06 and the 19-06-12.

ICSE '20, Seoul, Korea,
2020. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1: Rejected subforums with their properties

Subforum Number	Category Title	Subforum Title	Nb. Threads Updated	Subforum Nb. Posts Updated	Subforum Description
2	diabetes discussion	newly diagnosed	5243	72452	If you are newly diagnosed with diabetes and would like to share your experiences or support with other people that have recently been diagnosed, please use this forum.
19	off-topic	general chat	3940	128570	Talk about everything and anything; world events, what books you are reading or films you've seen, what the weather was like... it's up to you.
20	off-topic	forum games	62	151172	Post or join in with any forum games here!
21	off-topic	jokes and humour	2128	8579	The place for you to share those hilarious videos, pictures, stories and jokes!
39	diabetes news and research	diabetes news	4697	11999	Discuss all diabetes related news - from whether you agree with the news to how it may effect you.
40	diabetes news and research	diabetes research	207	1609	Discuss all aspects of diabetes research. Post links to diabetes research articles and their findings.
41	get involved	book and product reviews	49	549	Use this area of the community to review and discuss books and products. Peer review can help others to make the right choices.
42	get involved	community submitted guides & links	49	549	Use this area of the forum to submit guides and links that other members of the community may find helpful.
43	get involved	diabetes events	200	1132	List and discuss diabetes events in your area or the wider environment. If commercial in nature, please contact the moderating team before posting.

5.2. **Relevant Insights & Results**:
The scraping of all the data of the Diabetes.co.uk Forum [4] resulted in three datasets: *Subforums Dataset* see dataset 4.1, *Threads Dataset* see dataset 4.2 and *Posts Dataset* see dataset 4.3.
- *Subforums Dataset* see dataset 4.1 contains all the information specific to 43 subforums.
- *Threads Dataset* see dataset 4.2 contains all the information specific to the 113,361 threads.
- *Posts Dataset* see dataset 4.3 contains all the information specific to the 1,772,006 posts.

Note: There is relevant information about the *Posts Dataset* message property in the data process 8..

5.3. **Conclusion**: Three datasets that group all the information of Diabetes.co.uk Forum [4] have been created.

6. **Goal**: Create a sample of the original Diabetes.co.uk Forum [4] data.

6.1. **Procedure & Results**: Rejected 9 of the 43 subforums because they were out of topic or too specific to a subject and were not considered general conversation subforums see table 1.
Those subforums were rejected based on the Category Title, Subforum Title and Subforum Description properties.
Which led us with 34 subforums which contain 96,786 threads for a total of 1,395,395 posts see Table 2.
The selected and rejected subforum can be identified in the *Subforums Dataset* see dataset 4.1 with the attribute *subforum_is_selected*, 1 for selected and 0 for rejected.
Two percent of each remaining subforum's threads have been selected to create our sample dataset.
This sample dataset contains 1,921 threads for a total of 25,423 posts.
This sample dataset has been named the *Posts Sample Dataset* 4.7 see table 2.

6.2. **Conclusion**: The *Posts Sample Dataset* 4.7 containing two percent of the threads of each selected subforum has been created.

7. **Goal**: Parse the *Posts Sample Dataset* 4.7 into sentences.

7.1. **Procedure & Results**: For each post of each thread of the Posts Sample Dataset we splitted the post message into sentences using nltk sentence tokenizer.
After that the urls have been removed from the sentences.
Finally, all characters that were not alphanumeric characters or punctuation characters were removed from the sentence.
The parsing of the *Posts Sample Dataset* 4.7 led us with a dataset called *Sentences Dataset* 4.8.
This dataset contains 109,612 sentences.

7.2. **Conclusion**: The *Sentences Dataset* 4.8 containing all the sentences from the posts of *Posts Sample Dataset* 4.7 has been created.

8. **Goal**: Remove short sentences from *Sentences Dataset* 4.8.

8.1. **Procedure & Results**: Some of the sentences in *Sentences Dataset* 4.8 were really short. Examples: *Hi.*, *Thanks*, *Sorry.* etc.

Table 2: Selected subforums and Posts Sample Dataset properties

Subforum Number	Subforum Title	Subforum Nb. Threads	Subforum Nb. Posts	Posts Sample Dataset Nb. Threads	Posts Sample Dataset Nb. Posts
1	Greetings and Introductions	7655	73503	153	1496
3	Diabetes Discussions	12423	236927	248	3919
4	Ask a Question	27754	263314	555	5756
5	Type 1 Diabetes	12411	182400	248	5013
6	Insulin Pump Forum	3495	35900	69	646
7	Type 2 Diabetes	10014	242887	200	3141
8	Type 1.5/lada diabetes	436	7221	8	93
9	Type 3c (pancreatic) diabetes	95	1204	1	4
10	Prediabetes	1289	17549	25	186
11	Gestational diabetes	261	1763	5	33
12	Reactive hypoglycemia	344	7449	6	78
13	Diabetes soapbox - have your say	1280	25265	25	617
14	Success stories and testimonials	860	13196	17	223
15	Children & teens	551	5208	11	71
16	Young people/adults	227	2818	4	64
17	Parents	965	8806	19	217
18	Pregnancy	1005	10276	20	162
22	Food, nutrition and recipes	3431	40977	68	762
23	Low-carb diet forum	5132	116424	102	1368
24	Low calorie diets	284	6434	5	149
25	Vegetarian diet forum	82	3330	1	11
26	Gluten-free forum	30	354	0	0
27	Fasting	183	4666	3	229
28	Weight loss and dieting	726	12505	14	124
29	Diabetes medication and drugs	466	3857	9	76
30	Blood glucose monitoring	1963	24535	39	352
31	Diabetes complications	709	9043	14	102
32	Other health conditions and diabetes	606	7456	12	83
33	Emotional and mental health	219	3255	4	22
34	Alternative treatments	322	3314	6	94
35	Fitness, exercise and sport	843	14675	16	156
36	Jobs and employment	308	3776	6	57
37	Benefits	133	1902	2	23
38	Driving and dvla	284	3206	5	96

The sentences matching at least one of the criteria below were removed from the *Sentences Dataset* 4.8.
- Sentences with zero, one or two alnum characters
- Sentences that have only one words and no question mark character ("?")
- Sentences that are equal to the string "Click to expand"
4,604 sentences have been removed and 105,008 sentences have been kept.
Regarding: "Sentences that are equal to the string "Click to expand"":
All the text data of all posts has been retrieved.
The reason I could only get the data with "Click to expand..." in it is because this string is inserted in the HTML quotes's div and is removed with javascript if it does not need to be displayed. So basically, "Click to expand..." is in all post's text that have a quote in it.
Here is an example of the parsing of a sentence that contains "Click to expand...".
Input sentence: "Redsnapper said: We have an excellent steakhouse near us who offer a "blue cheese sauce" with steaks and their own in house burgers.Its fantastic and probably easy to make. Click to expand..."
Parsed to sentences:
Sentence 1: "Redsnapper said: We have an excellent steakhouse near us who offer a "blue cheese sauce" with steaks and their own in house burgers.Its fantastic and probably easy to make."
Sentence 2: "Click to expand..."
My script then removes the sentences that have "Click to expand..." in it.
Note: If the sentence before "Click to expand..." has wrong punctuation (like a "," instead of a "." in the example below) nltk parser will parse this string as a hole sentence.
Then with the script this sentence will be removed because it contains "Click to expand..."
Example sentence : "brassyblonde900 said: If after a reasonable time if you get no takers, flog it on eBay, Click to expand... xfieldok said: That's a thought, thanks."

8.2. **Conclusion**: Short sentences have been removed from the *Sentences Dataset* 4.10.

9. **Goal**: Remove all introduction sentences from *Sentences Dataset* 4.10.

9.1. **Procedure & Results**: Some of the threads in *Sentences Dataset* 4.10 were simply introduction thread. The threads were considered introduction thread by analyzing the thread's title. Examples: *Well am a new member from West Africa*, *New here*, *Hello!* were considered introduction threads.
By analyzing the thread's title of *Sentences Dataset* 4.10 manually a list of introduction keywords has been created see 3.
All threads with a thread's title that contained at least one introduction keywords were removed from *Sentences Dataset*.
The removed thread can be found in dataset 4.11. The remaining threads can be found in dataset 4.10.
3542 sentences have been removed and 101,466 sentences have been kept.

9.2. **Conclusion**: Sentences that were part of an introduction thread have been removed from the *Sentences Dataset* 4.12.

10. **Goal**: Identify interrogative sentences of *Sentences Dataset* 4.12.

Table 3: List of Introduction Keywords

Introduction Keywords
hello
hi
newbie
new here
new to forum
just joined
new member
hey
new forum member

10.1. **Procedure & Results**:
Our goal was to distinguish interrogative sentences from the non-interrogative sentences. To do so, the Stanford CoreNLP parser [6] has been used.
The clause level tags of the parsed sentences have been used see table 4.
Note: the Stanford CoreNLP uses Penn Treebank II Constituent Tags[7].
The sentences with a clause level tag equal to either *SBARQ* or *SQ* were classified as interrogative sentences.
3,569 sentences were classified as interrogative sentences.
This new dataset has been named *Sentences Annotated Dataset* 4.15.

10.2. **Conclusion**: Interrogative sentences have been categorized in *Sentences Annotated Dataset* 4.15.

11. **Goal**: Identify sentences with need keywords in *Sentences Annotated Dataset* 4.15.

11.1. **Procedure & Results**: The sentences of *Sentences Annotated Dataset* 4.15 that contain at least one need keywords of the *List of need keywords* presented in data process 4. were categorized as sentences with need keywords.
50,096 sentences were classified as sentences that contain need keywords.

11.2. **Conclusion**: Sentences that contain need keywords have been categorized in *Sentences Annotated Dataset* 4.16.

Table 4: Clause Tags Description

S	Simple declarative clause, i.e. one that is not introduced by a (possible empty) subordinating conjunction or a wh-word and that does not exhibit subject-verb inversion.
SBAR	Clause introduced by a (possibly empty) subordinating conjunction.
SBARQ	Direct question introduced by a wh-word or a wh-phrase. Indirect questions and relative clauses should be bracketed as SBAR, not SBARQ.
SINV	Inverted declarative sentence, i.e. one in which the subject follows the tensed verb or modal.
SQ	Inverted yes/no question, or main clause of a wh-question, following the wh-phrase in SBARQ.

12. **Goal**: Create a sample dataset from *Sentences Annotated Dataset* 4.16 to get it labelled with amazon mechanical turk.
    12.1. **Procedure & Results**:
    Three subdatasets of *Sentences Annotated Dataset* 4.16 have been created:
    - *All Question sentences Dataset* 4.17: contains 3569 sentences.
    - *All Sentences With Need Keywords That Are Not Question Sentences* 4.18: contains 47,775 sentences.
    - *All Other Sentences Dataset* 4.19: contains 50,123 sentences.

    3500 sentences of each subdataset have been randomly selected for a total of 10,500 sentences.
    This new dataset has been named *Mturk Sentences Sample Dataset* 4.20. Those 10,500 sentences comes from 1614 different threads.
    12.2. **Conclusion**: The dataset *Mturk Sentences Sample Dataset* 4.20 has been created from three subdataset of *Sentences Annotated Dataset* 4.16.
13. **Goal**: Find the best summarization technique for the thread's text.
    13.1. **Procedure**:
    All the posts from all the threads that have at least one sentence in *Mturk Sentences Sample Dataset no1* were grouped together to get the thread discussion as a hole text which gave us the *Thread's Text Dataset* 4.34. The *Thread's Text Dataset* 4.34 contains 1614 different threads.
    The thread's text has been retrieved to compute summarization on it.
    For all the threads in *Thread's Text Dataset* 4.34 summaries of 1 to 50 sentences have been computed from the thread's text.
    Three groups of summaries of 1 to 50 sentences have been produced using three different algorithms: *SumBasic* using sumy summarizer [2], *LexRank* using sumy summarizer [2] and *TextRank* using Gensim summarizer [9].
    Those three groups of summaries resulted in three new datasets respectively named *Threads Summarized With Sumbasic Dataset* 4.36, *Threads Summarized With LexRank Dataset* 4.35 and *Threads Summarized With TextRank Dataset* 4.37.
    All the summaries of 1 to 50 sentences were then evaluated with ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-4.
    13.2. **Relevant Insights & Results**:
    To evaluate the best summarization technique, the average ROUGE score for each summaries of different length has been calculated. The Figure 1 presents the average ROUGE-2 score in function of the number of sentences in the summary for the three summarization techniques we tested.
    The Figure 1 shows that LexRank summarization algorithm is best suited to summarize thread's text.
    It is important to note that only one graph of the ROUGE scores results is presented here because all the ROUGE scores results concluded the same thing.
    Note: There can be duplicate sentences in the summary produced by the summarization techniques that we used if the sentence is repeated multiple times in the original text. LexRank algorithm prioritizes sentences that are different from one another, but there can still be duplicate sentences in the summaries.

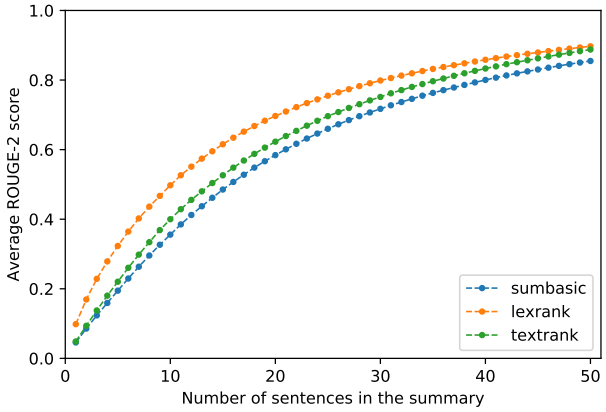

Figure 1: Average ROUGE-2 scores in function of the number of sentences in the summary

For the 1614 summaries of 20 sentences (so a bit less than 32,280 sentences) there are 100 sentences that comes at least more than once in the summary. As we discussed in the beginning of the project, the text data of each posts contains the post's author's text but also the quote's text (the quote's text is text that the author cited of another author).
For example, we might have this string for a post text's data: "catherinecherub said: Hi @Sean01 and welcome. Can you clarify your status as your information page says that you are Type1.5 Click to expand... Yes - but only for body building competitions. I only competed twice - I picked my moment to shine. It worked!"
So the quote can be identified with "text_cited_author said:" (in the example above text_cited_author = catherinecherub) and wrapped up with "Click to expand..."
We wanted to keep this information in the post text's data because it brings relevant information about what the author of the post is talking about.
So when a sentence is cited multiple times it can be found multiple times in the thread's text (the thread's text contains all the post's text).
    13.3. **Conclusion**: The dataset *Thread's Text Dataset* 4.34 has been created from the threads that have at least one sentence in *Mturk Sentences Sample Dataset no1*. Three thread's text summarized datasets have been created using three different summarization algorithm; Sumbasic, LexRank and TextRank. After analyzing the performances of the different summarization techniques, LexRank algorithm has been determined as most suited to summarize the thread's text of *Thread's Text Dataset* 4.34.
14. **Goal**: Find the best summary length.
    14.1. **Procedure**:
    The selection of the summary length has been based on the ROUGE scores presented in Figure 1. Summaries of 20 sentences have been selected because of their average ROUGE-2 score of ~ 0.75.

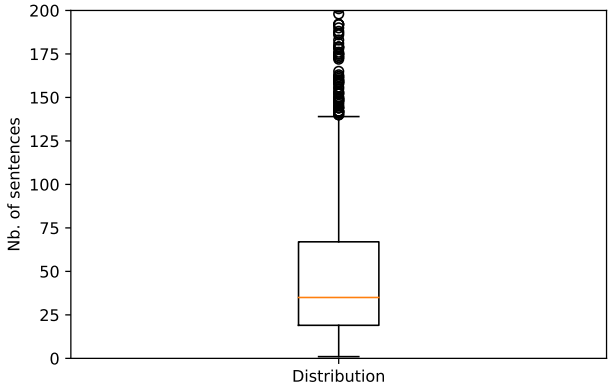

Figure 2: Distribution of the number of sentences per thread in *Sentences Dataset*

The 20 sentences summaries of *Threads Summarized With LexRank Dataset* will be named as *20 Sentences LexRank Summaries Dataset.*
Note: we should review why we selected 20 sentences. After making more research I can affirm that a ROUGE score of ~ 0.75 is really high.

14.2. **Relevant Insights & Results**:
The *Sentences Dataset* 4.12 has been used to evaluate the average number of sentences per thread. The average number of sentences per thread is 61±165 sentences. More details about the distribution of this result are presented in the Figure 2.
The Figure 3 presents more information about the frequency of the number of sentences per thread.

14.3. **Conclusion**: Summaries of 20 sentences have been selected based on the ROUGE scores presented in Figure 1 to form the *20 Sentences LexRank Summaries Dataset.*

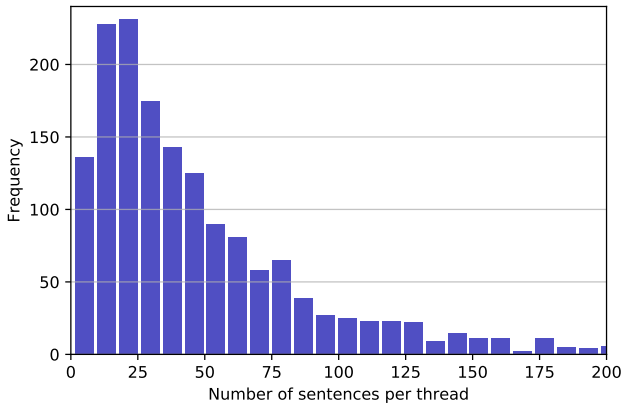

Figure 3: Frequency of the number of sentences per thread in *Sentences Dataset*

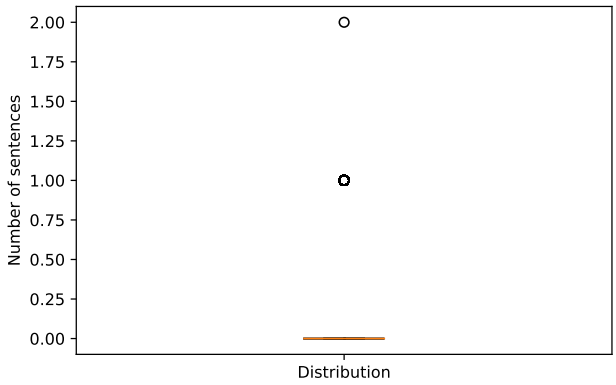

Figure 4: Distribution of the number of sentences per thread in the *20 Sentences LexRank Summaries Dataset* that are in *Mturk Sentences Sample Dataset*

15. **Goal**: Find links between the *20 Sentences LexRank Summaries Dataset* and the *Mturk Sentences Sample Dataset.*

15.1. **Procedure**:
The *20 Sentences LexRank Summaries Dataset* and *Mturk Sentences Sample Dataset* have been analyzed and statistics have been retrieved in order to find links between the two datasets.

15.2. **Relevant Insights & Results**:
The average number of sentences per thread of *20 Sentences LexRank Summaries Dataset* that are in *Mturk Sentences Sample Dataset* is 0.155514±0.364213 sentences per thread. More details about the distribution of this result are presented in the Figure 4.
There are 251 sentences that are in *20 Sentences LexRank Summaries Dataset* and also in *Mturk Sentences Sample Dataset.*
There are 250 out of the 1614 threads summaries that have at least one sentence in *Mturk Sentences Sample Dataset.*
The Figure 5 presents more information about the the average number of sentences per thread of *20 Sentences LexRank Summaries Dataset* that are in *Mturk Sentences Sample Dataset.*
The average number of words in the sentences of *Mturk Sentences Sample Dataset* is 17±9 words.
The average number of words in the sentences of *20 Sentences LexRank Summaries Dataset* is 24±10 words. More details about the distribution of those results are presented in the Figure 6.

15.3. **Conclusion**: The only link between the 20 sentences summaries of *20 Sentences LexRank Summaries Dataset* and *Mturk Sentences Sample Dataset* is that a small part of the sentences that are in the summaries are also in the *Mturk Sentences Sample Dataset.*

16. **Goal**: Create a dataset with all the first posts of all the threads.

16.1. **Procedure & Results**:
Selected all the first posts in the *Posts Dataset* see dataset 4.3 to create the *First Posts Dataset.*

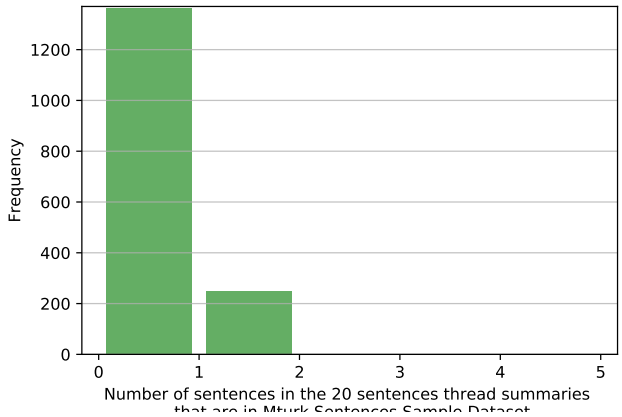

Figure 5: Frequency of the number of sentences in the *20 Sentences LexRank Summaries Dataset* that are in the *Mturk Sentences Sample Dataset*

16.2. **Conclusion**: The *First Posts Dataset* has been created from the *Posts Dataset* see dataset 4.3.

17. **Goal**: Analyze the labelled dataset of *Mturk Sentences Sample Dataset*.

17.1. **Procedure & Results**:
The dataset *Mturk Sentences Sample Dataset* has been labelled which gave us the *Mturk Sentences Sample Labelled Dataset* see dataset 4.23
Multiple datasets have been created to be able to analyse the data see datasets 4.24 4.25 4.26 4.27 4.28 4.29. For more details see the description of those datasets in Section 4.
Three workers labelled each of the 10,500 sentences of *Mturk Sentences Sample Dataset*. The distribution of those answers is presented in Table 5.

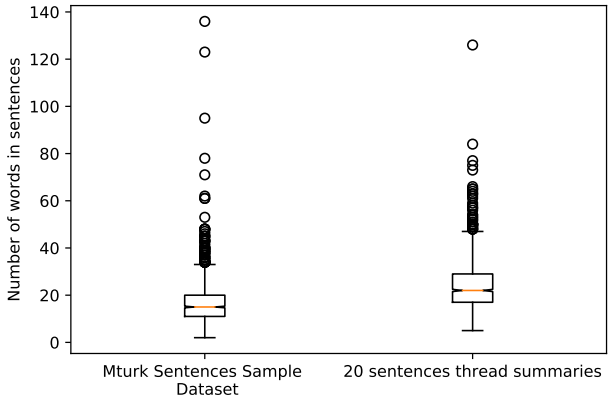

Figure 6: Number of words in sentences for *Mturk Sentences Sample Dataset* and the *20 Sentences LexRank Summaries Dataset*

Table 5: Distribution of workers answers in *Mturk Sentences Sample Labelled Dataset*

Label	Answers distribution	Number of answers	Totals	
1	yes yes yes	550	2118	10012
2	yes yes x	1568		
3	no no no	4430	7894	
4	no no x	3464		
5	cd cd cd	2	488	
6	cd cd x	62		
7	yes no cd	424		
Total workers all agreed (1,3,5)		4982		
Total workers don't all agreed (2,4,6,7)		5518		
Where "x": is any value				

The number of sentences with each sentence properties is presented in Table 6.
The distribution of the number of sentences with the properties is_question and expresses_a_need is presented in Table 7.
The distribution of the number of sentences with the properties has_annotations and expresses_a_need is presented in Table 8.
The distribution of the number of sentences with the properties is_question, has_annotations and expresses_a_need is presented in Table 9.

17.2. **Conclusion**: The *Mturk Sentences Sample Labelled Dataset* has been created and multiple datasets have been produced to be able to analyze it's data.

18. **Goal**: Create a classifier based on the labelled data.

Table 6: Number of sentences with each properties in *Mturk Sentences Sample Labelled Dataset*

is a question	3200	10012
is question	6812	
has annotations	5457	10012
has no annotations	4555	
expresses a need	2118	10012
does not express a need	7894	

Table 7: Distribution of sentences with the properties is_question and expresses_a_need in *Mturk Sentences Sample Labelled Dataset*

Sentence	expresses a need	does not express a need	
is a question	1546	1654	**3200**
is not a question	572	6240	**6812**
	2118	**7894**	**10012**

Table 8: Distribution of sentences with the properties has_annotations and expresses_a_need in *Mturk Sentences Sample Labelled Dataset*

Sentence	expresses a need	does not express a need	
has annotations	1452	4005	**5457**
has no annotations	666	3889	**4555**
	2118	**7894**	**10012**

18.1. **Procedure & Results**:
Two Random Forest Classifier have been created based on the *Mturk Sentences Sample Labelled Dataset.*
The First Random Forest Classifier uses BOW text representation see results in Table 10.
The Second Random Forest Classifier uses TF-IDF text representation see results in Table 11.
Two features have been added to the initial text feature; the sentence properties *is_question* and *has_annotations* :
See the results for the First Random Forest Classifier with the new features added in Table 12.
See the results for the Second Random Forest Classifier with the new features added in Table 13.

18.2. **Conclusion**: A Random Forest Classifier using BOW and TF-IDF text representation have been created. The scores of both of those implementation have been calculated.

Table 9: Distribution of sentences with the properties is_question, has_annotations and expresses_a_need in *Mturk Sentences Sample Labelled Dataset*

Sentence	expresses a need	does not express a need	
is not a question & has no annotations	150	3277	**3427**
is not a question & has annotations	422	2963	**3385**
is a question & has no annotations	516	612	**1128**
is a question & has annotations	1030	1042	**2072**
	2118	**7894**	**10012**

Table 10: Results of Random Forest Classifier with BOW

score	mean	std
accuracy	0.5408	0.1303
precision	0.5013	0.4987
recall	0.3276	0.3240
f1	0.3089	0.3200

Table 11: Results of Random Forest Classifier with TF-IDF

score	mean	std
accuracy	0.5575	0.1429
precision	0.5006	0.4993
recall	0.2776	0.2472
f1	0.3069	0.3224

Table 12: Results of Random Forest Classifier with BOW, is_question, and has_annotations

score	mean	std
accuracy	0.6220	0.2077
precision	0.5000	0.5000
recall	0.2688	0.3285
f1	0.3228	0.3900

Table 13: Results of Random Forest Classifier with TF-IDF, is_question, and has_annotations

score	mean	std
accuracy	0.6409	0.2118
precision	0.5000	0.5000
recall	0.2841	0.3461
f1	0.3351	0.3994

4 DATASETS

The groups of datasets are listed below:

- **Source Datasets**
- **Annotations Datasets**
- **Sentence Datasets**
- **Summary Datasets**
- **Analyze of Best Summarization Technique Datasets**
- **Classifier Results Datasets**

Source Datasets

The Diabeties.co.uk source datasets have been scrapped using scrapy library [10].The source code for scrapping the forums is detailed in the files under the folder:

`jh-summer19\Exercices\Exercices11_Extract_Threads\-\scrapyDiabetesForum\tutorial\spiders`

4.1 subforum_updated_classified.csv (*Subforums Dataset*)

The subforum_updated_classified.csv dataset contains all the properties listed below:

- subforum_number
- category_title
- category_link
- subforum_title
- subforum_is_selected
- subforum_link
- subforum_description
- subforum_nb_threads
- subforum_nb_posts
- nb_threads_updated
- subforum_nb_posts_updated

4.2 threads.csv (*Threads Dataset*)

The threads.csv dataset contains all the properties listed below:

- thread_id
- thread_title
- thread_link
- thread_author
- thread_startDate
- thread_replies
- thread_views

4.3 Posts data: all i.csv where i is in [1, 43] (*Posts Dataset*)

The posts data which are all in the i.csv datasets contains all the properties listed below:

- thread_id
- thread_title
- thread_link
- thread_author
- thread_startDate
- thread_replies
- thread_views
- post_id
- post_username
- post_userTitle
- post_userNbMessages
- post_userLikesReceived
- post_userTrophyPoints
- post_DateTime
- post_like
- post_agree
- post_useful
- post_funny
- post_informative
- post_friendly
- post_optimistic
- post_hug
- post_number
- post_messageText
- post_quoteText
- subforum_number

Annotations Datasets

The annocations source's data has been extracted manually directly from the Diabetes.co.uk Forum [4] online.

Note: Here annotations is used as a synonym for keywords. So a "need keyword" is equivalent to "annotation". The term "annotation" has been used in this section to stay consistent with the dataset's name.

4.4 annotations_source.csv

All the annotations in dataset 4.4 have been extracted an grouped together to form the dataset 4.5

4.5 annotations.csv

All the synonyms of the annotations in dataset 4.5 have been found using NLTK and added to it to create the dataset 4.6 presented below:

(see get_synonyms.py)

4.6 annotations_with_synonyms.csv

Sentence Datasets

The history of the **Sentence Datasets** is presented in the Figure 8.

The file below contains the sample of threads extracted from the selected subforums (2% of threads of each selected subforums) (see dataset_builder_get_percentage_of_threads_data .py)

4.7 0.02-of_threads_random_sample.csv (*Posts Sample Dataset*)

Took all the posts of 0.02-of_threads_random_sample.csv and parsed them into sentences (see parse_posts_to_sentences.py)

4.8 parsed_0.02_of_threads_to_sentences.csv (*Sentences Dataset*)

Contains all the sentences of the parsed posts with additional info (nb. of words/sentence & nb. of alnum char/sentence)

4.9 parsed_0.02_of_threads_to_sentences_with_stats.csv

Removed all the sentences from
parsed_0.02_of_threads_to_sentences_with_stats.csv

- with 0, 1 or 2 alnum characters
- that have 1 words and no question mark (?)
- that are matching one of the short_sentences_to_remove (see below)

(see remove_short_messages.py)
Which gave us the two files below:

4.10 parsed_0.02_of_threads_to_sentences_kept_sentences.csv (*Sentences Dataset* updated)

4.11 5. parsed_0.02_of_threads_to_sentences_removed_sentences.csv

After that, from
parsed_0.02_of_threads_to_sentences_kept_sentences.csv,
we removed the introduction threads, threads that have a title starting with Hi, Hello, New here etc. Note: those files are poorly named, we are not removing threads, but sentences.
(see remove_introduction_threads.py)
Which gave us the two files below:

4.12 parsed_0.02_of_threads_kept_threads.csv (*Sentences Dataset* updated)

4.13 parsed_0.02_of_threads_removed_threads.csv

Tagged sentences of parsed_0.02_of_threads_kept_threads.csv with core_nlp_clause_tag with core_nlp parser
(see core_nlp_interrogative_sentence_tagger.py)
Which gave us the file below:

4.14 parsed_0.02_kept_threads_with_core_nlp_clause_tags.csv

Tagged sentences of
parsed_0.02_kept_threads_with_core_nlp_clause_tags.csv
with is_question based on the core_nlp_clause_tags:
(see core_nlp_clause_tags_to_is_question.py)
Which gave us the file below:

4.15 parsed_0.02_kept_threads_with_is_question.csv (*Sentences Annotated Dataset*)

Tagged sentences of parsed_0.02_kept_threads_with_is_question.csv with has_annotations with keywords parser
(see annotations_tagger.py)
Which gave us the file below:

4.16 parsed_0.02_kept_threads_with_is_question_and_annotations.csv (*Sentences Annotated Dataset* updated)

Grouped the data into three sub-datasets:

- Dataset 1: question sentences (is_question tag)
- Dataset 2: sentences that contains annotations and are not question sentences (!is_question and has_annotations tag)
- Dataset 3: sentences that do not contains annotations and are not question sentences (!is_question and !has_annotations tag)

(see sampler_for_mturk_no1.py)
Which gave us the files below:

4.17 all_question_sentences_mturk_no2.csv

4.18 all_annotations_without_questions_sentences_mturk_no2.csv

4.19 all_other_sentences_mturk_no2.csv

Took three random sample from
parsed_0.02_kept_threads_with_is_question_and_annotations.csv

- Sample dataset #1: 3500 question sentences (is_question tag)
- Sample dataset #2: 3500 sentences that contains annotations and are not question sentences (!is_question and has_annotations tag)
- Sample dataset #3: 3500 sentences that do not contains annotations and are not question sentences (!is_question and !has_annotations tag)

(see sampler_for_mturk_no1.py)
Which gave us the file below:

4.20 sample_dataset_mturk_no2.csv (*Mturk Sentences Sample Dataset*)

Added sentence_id to each sentence of sample_dataset_mturk_no2.csv for submitting to amazon mechanical turk (mturk).
(see add_sentence_id_to_dataset.py)
Which gave us the file below:

4.21 sample_dataset_mturk_no2_with_sentence_ids.csv

Grouped 5 sentences per row to submit multiple sentences in one HIT in mturk.
(see group_n_sentences_in_one_row_for_mturk.py)
Which gave us the file below:

4.22 sample_dataset_mturk_no2_group_of_5_sent.csv

Retrieved labelled data from mturk

4.23 Batch_3719368_batch_results

Separated the 5 sentences per row in one sentence per row.
(sample_mturk_no2_results_to_rows.py)
Which gave us the file below:

4.24 sample_dataset_mturk_no2_results_to_rows.csv

The sentence property from sample_dataset_mturk_no2_with_sentence_ids.csv has been retrieved with the sentence_id property and added to sample_dataset_mturk_no2_results_to_rows.csv.

(sample_mturk_no2_results_to_rows_with_sentences.py)

Which gave us the file below:

4.25 sample_dataset_mturk_no2_results_to_rows_with_sentence.csv

The value with "invalid_value" in property "expresses_a_need" from dataset sample_dataset_mturk_no2_results_to_rows_with_sentence.csv were manually labelled

(no python file)

Which gave us the file below:

4.26 sample_dataset_mturk_no2_results_to_rows_with_sentence_and_invalid_values_labelled.csv

Mapped the 3 worker's answers to 7 different groups of answers see list below:

Answers to Group Label:

- 1 : yes yes yes
- 2 : yes yes x
- 3 : no no no
- 4 : no no x
- 5 : cd cd cd
- 6 : cd cd x
- 7 : yes no cd

Where x is any value

So

Group Label 1, 3, 5 are answers where all workers agreed on the label

Group Label 2, 4, 6, 7 are answers where workers disagreed on the label

(sample_mturk_no2_results_grouped.py)

Which gave us the file below:

4.27 sample_mturk_no2_results_grouped.csv

All the properties from sample_dataset_mturk_no2_with_sentence_ids.csv have been retrieved with the sentence_id property and added to sample_mturk_no2_results_grouped.csv.

(sample_mturk_no2_final_results_merged.py)

Which gave us the file below:

4.28 sample_dataset_mturk_no2_merged_results.csv

Labelled worker's answer to expresses_a_need_final property based on the expresses_a_need_grouped property

New label:

- expresses_a_need_grouped: 1 2 are mapped to expresses_a_need_final: 1

- expresses_a_need_grouped: 3 4 are mapped to expresses_a_need_final: 2

(dataset_builder_for_classifier.py)

Which gave us the file below:

4.29 dataset_for_classifier.csv

Summary Datasets

For each thread_ids selected in the sample_dataset_mturk_no2.csv we retrieved all the thread conversation from

0.02-of_threads_random_sample.csv.

(see get_datasets_for_summarization.py)

Which gave us the file below:

4.30 threads_text_for_summarization_no2.csv (*Thread's Text Dataset* for no2 datasets)

For each thread we summarized the hole thread conversation for 20 sentences.

(thread_text in threads_text_for_summarization_no2.csv)

Which gave us the file below:

4.31 threads_summarized_lexrank_20_to_20.csv (*20 Sentences LexRank Summaries Dataset*)

All the text of the summaries of *20 Sentences LexRank Summaries Dataset* has been parsed to sentences.

(summaries_to_sentences.py)

Which gave us the file below:

4.32 threads_summarized_lexrank_20_to_20_no_text_to_sentences.csv

A sample of 3500 sentences has been retrieved from threads_summarized_lexrank_20_to_20_no_text_to_sentences.csv.

(sampler_for_summaries.py)

Which gave us the file below:

4.33 sample_summary_sentence_no1.csv

Analyze of Best Summarization Technique Datasets

The history of the **Analyze of Best Summarization Technique Datasets** is presented in the Figure 7.

Note: The analysis of the best summarization technique has been done with the threads_text_for_summarization_no1.py which is based on sample_dataset_mturk_no1.csv.

The sentences that are in those two datasets have been parsed with the function remove_special_characters_no1, see code 1.

Listing 1: Remove Special Characters Function for no1 Datasets

```
# 06/04/19 sentence_parser.py
def remove_special_characters_no1(self, text):
    return re.sub(r"[^A-Za-z0-9(),.!?@\'\ \"\_\n]", "␣", text)
```

On the other hand the sentences that are in all the datasets prensented in the **Sentence Datasets** and **Summary Datasets** sections have been parsed with the function
remove_special_characters_no2, see code 2.

Listing 2: Remove Special Characters Function for no2 Datasets

```
# 07/15/19 sentence_parser.py
def remove_special_characters_no2(self, text):
    return re.sub(r"[^A-Za-z0-9(),.!\\?\'\ \"
\_\n\/#$%&*+\-:;<=>@\[\]^_{|}~]",
"␣", text)
```

This difference about the parsing of the two datasets can be explained because after analysis, we decided to keep all punctuation characters for the **Sentence Datasets** see listing 2 instead of just a few see listing 1. Due to time restrictions all the summary datasets presented in this section have not been regenerated with the function remove_special_characters_no2, see listing 2.

It is possible to affirm that all the information and the insights retrieved for finding the best summarization technique from the no1 datasets are as valid as if those datasets were parsed with the function remove_special_characters_no2, see listing 2 because all the summarization algorithms used have their own sentence parser that removes all punctuation characters. So even if we kept more punctuation character with remove_special_characters_no2 see listing 2 those would have been removed for the summary algorithm.

The datasets created to analyse the best summarization techniques are detailed below:

For each thread_ids selected in the
sample_dataset_mturk_no1.csv
we retrieved all the thread conversation from
0.02-of_threads_random_sample.csv.
(see get_datasets_for_summarization.py)
Which gave us the file below:

4.34 threads_text_for_summarization_no1.csv (*Thread's Text Dataset* for no1 datasets)

For each thread we summarized the hole thread conversation
(thread_text in threads_text_for_summarization_no1.csv)
for x sentences x going from 1 to 50 sentences. To generate those summaries different libraries have been used:

- **threads_summarized_sumbasic_1_to_50.csv**: uses sumy library [2]
- **threads_summarized_lexrank_1_to_50.csv**: uses sumy library [2]
- **threads_summarized_textrank_1_to_50.csv**: uses Gensim library [9]
  (see: https://github.com/HuppeJ/gensim for a faster version for getting multiple summaries of different length at once.)
  (see thread_summarizer.py)
  Which gave us the files below:

4.35 threads_summarized_lexrank_1_to_50.csv (*Threads Summarized With LexRank Dataset* for no1 datasets)

4.36 threads_summarized_sumbasic_1_to_50.csv (*Threads Summarized With Sumbasic Dataset* for no1 datasets)

4.37 threads_summarized_textrank_1_to_50.csv (*Threads Summarized With TextRank Dataset* for no1 datasets)

For clarity reason we removed the thread_text property in the files above so that the file could be opened in excel and easily visualized.
(see pd_tools_drop_column.py)
Which gave us the files below:

4.38 threads_summarized_lexrank-_1_to_50_no_text.csv, threads_summarized_sumbasic-_1_to_50_no_text.csv & threads_summarized_textrank-_1_to_50_no_text.csv

Based on threads_summarized_X_1_to_50.csv files the rouge-1, rouge-2, rouge-3 and rouge-4 scores has been calculated:
(see thread_summary_rouge_scores.py)
Which gave us the files below:

4.39 threads_summarized_lexrank-_rouge_i_scores_1_to_50.csv, threads_summarized_sumbasic-_rouge_i_scores_1_to_50.csv & threads_summarized_textrank-_rouge_i_scores_1_to_50.csv

Classifier Results Datasets

All the datasets produced detailing the results of the Classifier can be found here *jh-summer19/project/rfut/data_output/mturk /submission_mturk_no2/results*

REFERENCES

[1] Trip Advisor. Trip Advisor. 2019. URL: https://www.tripadvisor.ca/ForumHome (visited on 06/24/2019).
[2] Mišo Belica. Sumy. 2019. URL: https://pypi.org/project/sumy/ (visited on 06/24/2019).
[3] Steven Bethard <Steven.Bethard@colorado.edu> Steven Bird <stevenbird1@gmail.com> Edward Loper <edloper@gmail.com> Nitin Madnani <nmadnani@ets.org> Nasruddin A'aidil Shari Sim Wei Ying Geraldine Soe Lynn Francis Bond <bond@ieee.org>. nltk.corpus.reader.wordnet. 2019. URL: https://www.nltk.org/_modules/nltk/corpus/reader/wordnet.html (visited on 06/24/2019).
[4] diabetes.co.uk. diabetes.co.uk. 2019. URL: https://www.diabetes.co.uk/forum/ (visited on 06/24/2019).
[5] Lexico.com. Lexico. 2019. URL: https://www.lexico.com/en (visited on 06/24/2019).
[6] Christopher D. Manning et al. "The Stanford CoreNLP Natural Language Processing Toolkit". In: Association for Computational Linguistics (ACL) System Demonstrations. 2014, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

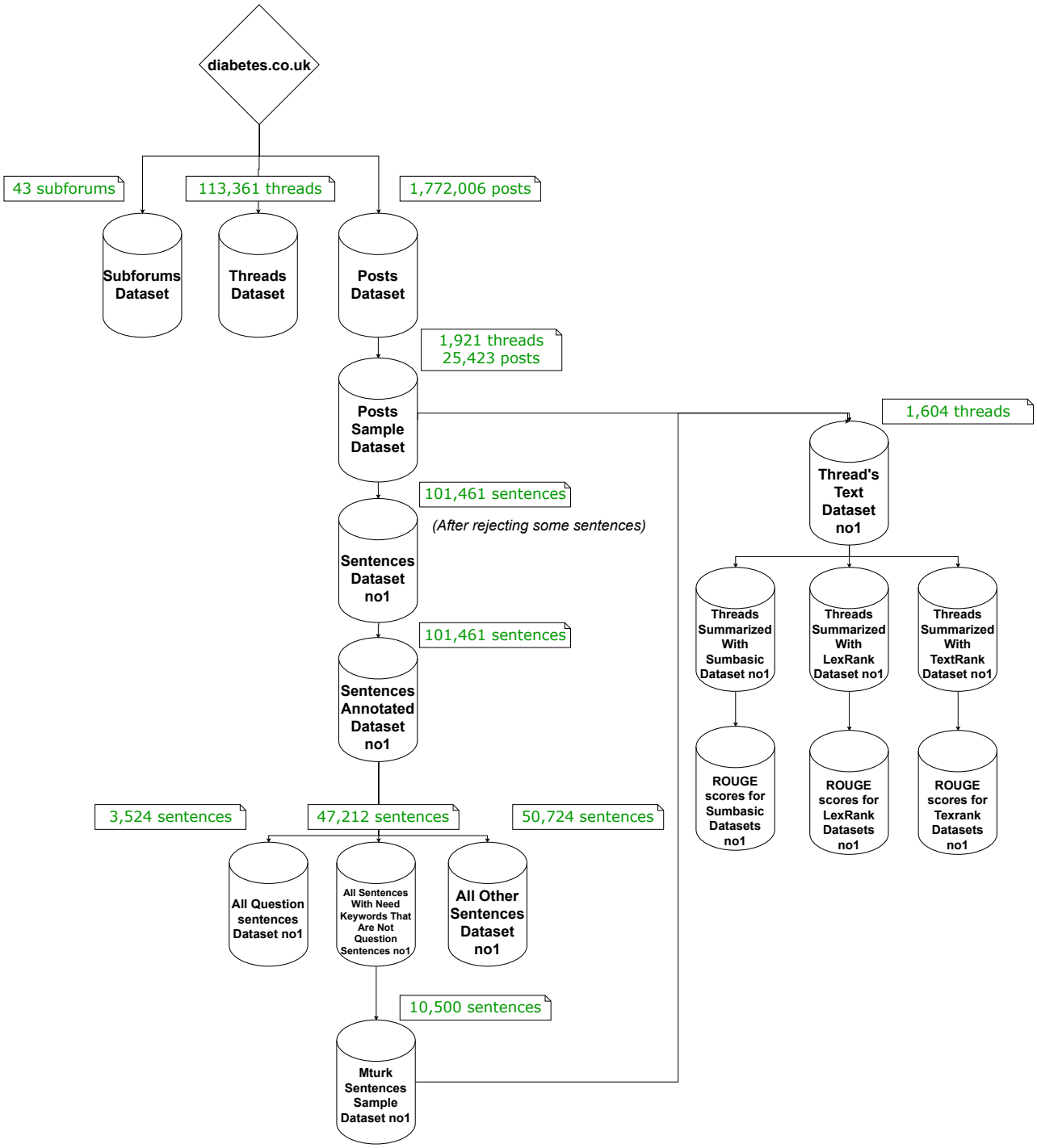

Figure 7: Datasets no1 History

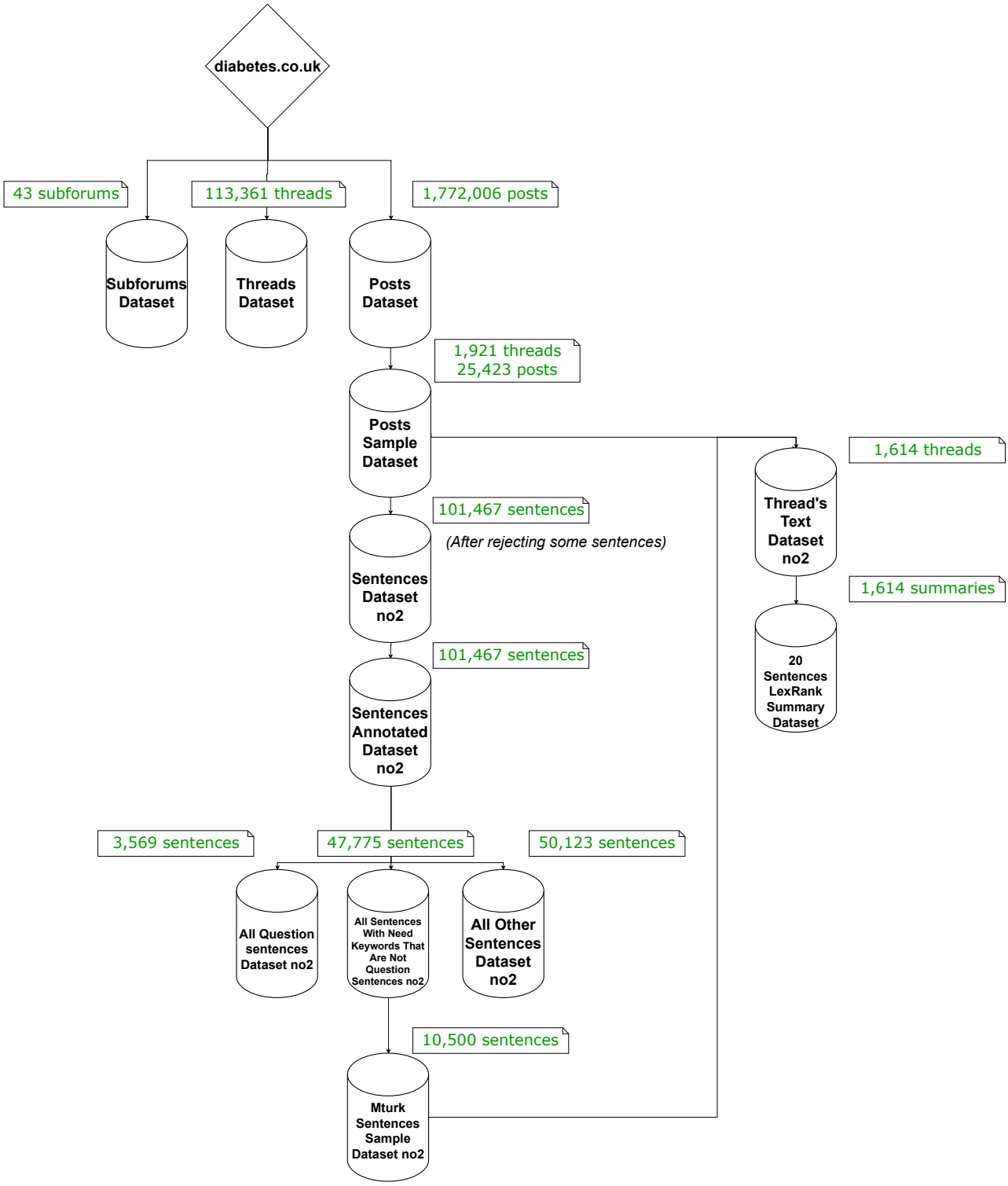

Figure 8: Datasets no2 History

[7] Mitchell Marcus et al. "The Penn Treebank: Annotating Predicate Argument Structure". In: Proceedings of the Workshop on Human Language Technology. HLT '94. Plainsboro, NJ: Association for Computational Linguistics, 1994, pp. 114–119. ISBN: 1-55860-357-3. DOI: 10.3115/1075812.1075835. URL: https://doi.org/10.3115/1075812.1075835.
[8] Reddit.com. Reddit. 2019. URL: https://www.reddit.com/ (visited on 06/24/2019).
[9] Radim Rehurek. Gensim. 2019. URL: https://pypi.org/project/gensim/ (visited on 06/24/2019).
[10] scrapy.org. Scrapy. 2019. URL: https://scrapy.org/ (visited on 06/24/2019).