# Project #1: A Multilingual Dialogue Dataset Technical Description of Corpus

### COMP551: Applied Machine Learning

**Team name: GreyJay**
Amanda Boatswain Jacques, ID: 260535594, amanda.boatswainjacques@mail.mcgill.ca
Taha Ghassemi, ID: 260516043, taha.ghassemi@mail.mcgill.ca
AhmadReza GodarvandChegini, ID: 260795422, ahmadreza.godarvandchegini@mail.mcgill.ca

McGill University
Wednesday, September 27, 2017

## I. INTRODUCTION

We assembled a corpus of conversations on the French-language health forum on Doctissimo.fr using data from the year of 2015.

**Forum URL:** http://forum.doctissimo.fr/
**Corpus URL:** https://drive.google.com/open?id=0B7z3HM2_6X-LVzVnT0hpRm5LaVE

## II. DATASET DESCRIPTION

***Website Content***

Our language corpus was created by compiling a set of French conversations retrieved from the medical forum Doctissimo. These conversations correspond to threads, posts, and discussions between users of the website over the two years. The Doctissimo forum is a website where thousands of users can discuss various health-related topics such as medication, nutrition, psychology, family, animals and more. Messages and replies on the forum consist primarily of people who are either seeking medical advice from online doctors, seeking opinions and recommendations from other users, or who would like to pursue conversations centered around a shared topic of interest. This allows for a certain degree of formality and minimal structure in the utterances of each user, and reduces the occasional collection of random and nonsensical conversation that is commonly found in online chat rooms. Moreover, the site has a directory of threads that are dated from the year 2000 to this very day, and some individual categories contain over 4 million user replies, showing potential for the creation of a dataset rich in diversity, length and complexity.

***Code Architecture***

To generate our corpus, we used a method known as web scraping. Web scraping is done by programmatically accessing a webpage, downloading it and extracting information from it [1].

A custom-made web crawler was designed for the Doctissimo site to fetch each page selected to create our corpus. The crawler follows all the links of the provided webpage until it has explored all the required links [2].

Our scraping software was built in the Python language (ver 3.0) using a library named Scrapy. First, we create an object of the scrapy.Spider class which will proceed to collect information from the entire forum. The forum is organized into individual topics which each correspond to a single URL or request. This master list is passed to a Spider which will download the corresponding webpage. A URL detection pattern for the webpage is created so the scraper can retrieve pages related to a specific category within a given year and month. Once the information is retrieved, we continue by parsing the content of the page, and then search for all recorded thread topics and the posts related to them [3].

The Scrapy architecture has multiple components that interact with each other to provide a steady and rapid flow of data. These components include the Scrapy Engine, the Scheduler, the Downloader, the Spider, the

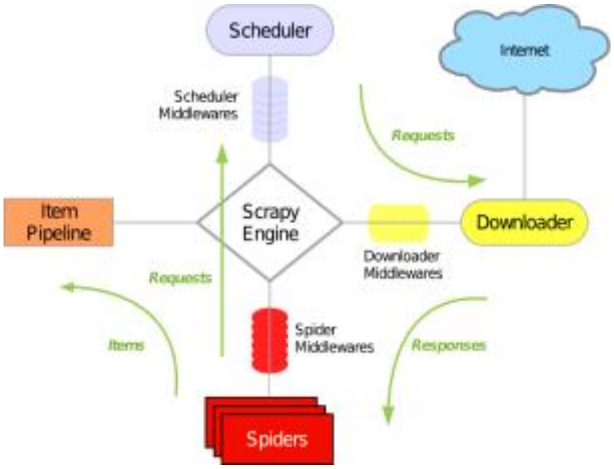

Fig. 1. Architecture of a Scrapy Engine. [3]

Item Pipeline. The Engine controls the overall data flow, and sends requests to the Scheduler which will enqueue these webpages before they are fed to the Downloader. The Downloader will then call on the Spider to parse and extract information from the webpage. Once this is completed, the data is sent to the Item Pipeline, which will format it and then save it. The process will continue until all the requests stored by the Scheduler have been scraped. Figure 1 represents a flow diagram of the Scrapy architecture [3].

After creating our Scrapy architecture, we proceeded by filtering the raw data from the webpages. The pages searched were dated from the entire year of 2015, some of the specifics of our corpus are reported in table 1.

| Doctissimo Corpus Composition | | | |
|---|---|---|---|
| **Utterances** | **Words** | **Turns** | **Conversations** |
| 291643 | 19643940 | 291479 | 9185 |

To guarantee an appropriate number of utterances per conversation, we limited our search to all threads that had at least 3 replies. Replies which consisted of images or “emojis” were removed to restrict our corpus to French text. As a result, comments which consisted of only these things were omitted.

Table 2. Characteristics of our Doctissimo Corpus.

Utterances were separated between users using a system of UUIDs or universally unique identifiers. These were 16-byte character strings which were derived using the SHA-1 hash function. This was done to guarantee anonymity between users while keeping them traceable. Thus, if a user posted a message in more than one thread, they would have the same UUID in the corpus [4]. Users which had a generic tag of “deleted account” were omitted from the dataset to avoid appearing to be the same person. Utterances that spanned over multiple paragraphs were separated with the help of a special token for paragraph breaks, <br>. This allowed our dataset to gain another interesting dimension for testing. By keeping the users retraceable, researches could study conversation patterns of various users (whether they utter one-line sentences, or prefer longer, more complete replies).

## III. DISCUSSION

Our corpus is primarily composed of French human-human conversations corresponding to casual spoken language. As stated by Serban et al., written and spoken language show difference in their individual structures [5]. Therefore, it has many unique properties such as the presence of abbreviations and slang. It is almost completely organic, with little processing performed on the content. The forum is mostly centered around medical advice, but there is a breath of topics, and conversations range from serious to lighthearted. This could be an appropriate dataset for training a model that would aim to recognize and mimic everyday French language. A full list of all the topics on the Doctissimo site can be found in Figure 3, and it is important to note that these topics are further divided into sub-topics or categories ranging from 10 to 100 per topic.

1. Health
2. Pregnancy and Babies
3. Fashion
4. Beauty
5. Nutrition
6. Psychology
7. Sexuality
8. Leisure
9. People
10. Medication
11. Fitness and Sports
12. Practical Life
13. Animals
14. Family
15. Cooking

Many existing French corpuses are composed of individual words, such as *Lexique3*, developed by the University of Savoie Mont-Blanc (150 000 words) [6]. Another French Corpus known as the *Corpus Frantext*, which was developed by the University of Nancy, contains 500 pieces of French literature from the 18th to 20th centuries [7]. Our corpus differs from these two for it is mostly composed of common French sentences, and it is more complex than individual words but less formal than literature.

Fig. 3. List of all the forum topics on the Doctissimo website.

## IV. STATEMENT OF CONTRIBUTIONS

Amanda Boatswain Jacques wrote a generous portion of the report and made minor contributions to the code. Mr. Godarzvand Chegini wrote most of the Python program. Mr. Ghassemi contributed to the code and writing/proofreading the report. We hereby state that all the work presented in this report is that of the authors.

## V. REFERENCES

1. Web Scraping. (n.d.). In Wikipedia. Retrieved September 23rd, 2017. From: https://en.wikipedia.org/wiki/-Web_scraping

2. Import.io. “How to Crawl a Website the Right Way.” Retrieved September 23rd, 2017. From: https://www.import.io/post/how-to-crawl-a-website-the-right-way/

3. Scrapy Documentation. “Architecture Structure.” Retrieved September 23rd, 2017. From: https://doc.scrapy.org/en/latest/topics/-architecture.html

4. The Python Standard Library. “ 20.15. uuid - UUID objects according to RFC 4122”. Retrieved September 24th, 2017. From:https://docs.python.org/2/-library/uuid.html

5. I. Serban, R. Lowe, P. Henderson, L. Charlin, J. Pineau. “A Survey of Available Corpora for Building Data-Driven Dialogue Systems.” Cornell University Library. arXiv:1512.05742 [cs.CL] March 2017.

6. Université de Savoie Mont-Blanc. “Lexique.” Retrieved September 27th, 2017, From: http://www.lexique.org/telLexique.php

7. Ortolang. “Corpus Frantext.” Retrieved September 27th, 2017. From: http://www.cnrtl.fr/corpus/frantext/

# Appendix: Code

```
import scrapy
import uuid
import re
from lxml import etree

class DoctissimoSpider(scrapy.Spider):
    name = "docSpidy"

    def start_requests(self):
        forums = ['http://forum.doctissimo.fr/top_topics/grossesse-
bebe/',
                  'http://forum.doctissimo.fr/top_topics/mode/',
                  'http://forum.doctissimo.fr/top_topics/forme-
beaute/',
                  'http://forum.doctissimo.fr/top_topics/nutrition/',
                  'http://forum.doctissimo.fr/top_topics/psychologie/',
                  'http://forum.doctissimo.fr/top_topics/doctissimo/',
                  'http://forum.doctissimo.fr/top_topics/loisirs/',
                  'http://forum.doctissimo.fr/top_topics/people-stars/',

'http://forum.doctissimo.fr/top_topics/medicaments/',
                  'http://forum.doctissimo.fr/top_topics/forme-sport/',
                  'http://forum.doctissimo.fr/top_topics/viepratique/',
                  'http://forum.doctissimo.fr/top_topics/animaux/',
                  'http://forum.doctissimo.fr/top_topics/famille/',
                  'http://forum.doctissimo.fr/top_topics/cuisine/',]
        for page in [s + str(year) + '/' + str(month).zfill(2) + '/' for
year in range(2015,2016) for month in range(1,13) for s in
forums]:
            yield scrapy.Request(page)



    def parse(self, response):
        def adeqReply(discussion):
            replies =
discussion.xpath('./td[@class="sujetCase7"]/text()').extract_first
()
            try:
                return int(replies) >= 3
            except:
                return True

        def undeletedOP(discussion):
            return
discussion.xpath('./td[contains(@class,"sujetCase6")]/text()').ext
ract_first() != "Profil supprimé"

        for discussion in
response.xpath('//*[@id="block_topics_list"]/tbody/tr'):
            if adeqReply(discussion) and undeletedOP(discussion):
                href =
discussion.xpath('./td[@class="sujetCase3"]/a/@href').extract_fi
rst()
                yield response.follow(href, self.discussionScraper)


    def discussionScraper(self, response):
        dialog = etree.Element("s")

        for post in
response.xpath('//div[@id="container"]/div[@class="containerf
orum"]/div[@class="content-du-
forum"]/div/div[@id="lesforums"]/div[@class="container"]/div
[@class="mesdiscussions"]/td/div[@id="topic"]/table[@id]/tr[1
]'):
            # List containing each username in the conv. as a string
            user =
post.xpath('./td[@class="messCase1"]/div[2]/b[@class="s2"]/de
scendant-or-self::*[last()]/text()').extract_first()
            # Make a UUID using a SHA-1 hash of a namespace
UUID and a name
            # uuid.NAMESPACE_DNS = When this namespace is
specified, the name string is a fully-qualified domain name
            uid = str(uuid.uuid5(uuid.NAMESPACE_DNS, user))

            # List containing each utterance in order of appearance
from the conv.
            utt =
post.xpath('./td[@class="messCase2"]/div[@class="post_conten
t"]/div[not(@class="edited") and not(@class="clear")]/text() |
./td[@class="messCase2"]/div[@class="post_content"]/div[not(
@class="edited") and not(@class="clear")]/p/text()').extract()
            if all([u.isspace() for u in utt]) or utt==[]:
                continue

            utt_node = etree.SubElement(dialog, 'utt', attrib={'uid' :
uid})

            for br in utt:
                if br.isspace() or br==[]:
                    continue
                paragraph = etree.SubElement(utt_node, 'br')
                paragraph.text = br

        with open('GreyJay_fre.xml', 'a', encoding='utf-8') as f:
            f.write(etree.tostring(dialog, pretty_print=True,
encoding='unicode'))
```