UNIVERSITÉ DE LORRAINE

L'INSTITUT DES SCIENCES DU DIGITAL MANAGEMENT COGNITION

Data Mining Project 2

Student:
Peter Pribil,Shalini Priya,Nasser

A final report for the Data Mining Project 2 course

Academic year 2021-2022

February 28, 2022

1 Introduction

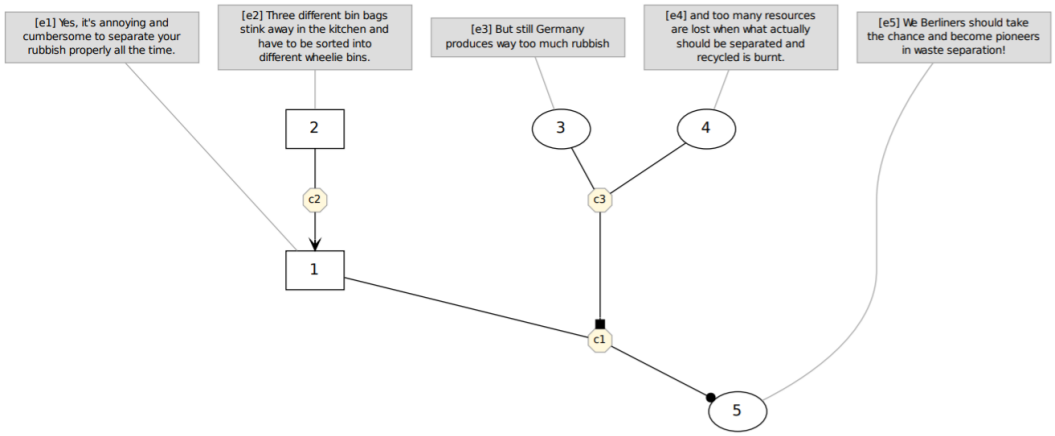
As part of the data mining project, were given the following to work on:

- Argument context for all the files
- Argument context for files with "pro"
- Argument context for files with "con"
- Microtexts as raw file, as PDF, and as xml with all the information
- Argument attribute patterns

The argument patterns are represented in the following way:

```
15: "t # 15 v 0 CC v 1 _ v 2 _ v 3 _ e 0 1 reb e 1 2 sup e 1 3 und "
```

This pattern, for example, can be found in the following microtext:



To interpret the results better, an xml file is provided for better visualization:

```
<adu id="a1" type="opp"/>
<adu id="a2" type="opp"/>
<adu id="a3" type="pro"/>
<adu id="a4" type="pro"/>
<adu id="a5" type="pro"/>
<edge id="c6" src="e1" trg="a1" type="seg"/>
<edge id="c7" src="e2" trg="a2" type="seg"/>
<edge id="c8" src="e3" trg="a3" type="seg"/>
<edge id="c9" src="e4" trg="a4" type="seg"/>
<edge id="c10" src="e5" trg="a5" type="seg"/>
<edge id="c1" src="a1" trg="a5" type="reb"/>
<edge id="c2" src="a2" trg="a1" type="sup"/>
<edge id="c3" src="a3" trg="c1" type="und"/>
<edge id="c4" src="a4" trg="c3" type="add"/>
```

To explain, from top-to-bottom:
We have five sentences (e1, e2, e3, e4, e5).
These are connected to their own units in either squares or circles (a1, a2, a3...)
These represent "pro" or "opp", as in pro statement, or opposite statement.
The edges(c6, c7, c8) connect these two, as you can see in, for example, id="c6" is connected through "e1", the source, to "a1", the target.

Then, we have the edges visible on the picture (c1, c2, c3, c4). These connect the "bubbles" or "squares" to each other. For example, "c1" connects "a1" and "a5", and the type "reb" indicates disagreement between "a1" and "a5".
We also have the type "sup", which indicates support, and we have that from "a2" to "a1". Both of them are in square circles, meaning they are "opp", so "a2" is supporting "a1".
Then we have type "und", which is directly connecting "a3" to the edge "c1". This means "a3" by itself is incomplete, and needs to join to a previous discourse element.
And last, but not least, we have the type "add", which we observe between "a4" and "c3". This means it's an addition to the previous sentence, "a3". "a3" is linked to "c1" directly through "c3", so "a4" is there as "addition" to "a3" on its way to "c1".

There are additional types of relations, less and more complicated structures, but for the sake of brevity, we've covered something that's understood fast.

The goal of the project is to analyze these microtexts based on their argument structures, and see whether there are any patterns from which we could infer some kind of a classification to decide whether a microtext is for or against a given statement.

2 Algorithms

We have implemented two algorithms :

- FP-Growth
- gSpan

FP-Growth is an unsupervised machine learning technique used for association rule mining which is faster than apriori.A frequent pattern is generated without the need for candidate generation.FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FP tree.

This tree structure maintains the association between the itemsets.The database is fragmented using one frequent item. This fragmented part is called "pattern fragment".The itemsets of these fragmented patterns are analyzed. Thus with this method, the search for frequent itemsets is reduced comparatively.This algorithm has been used in our project to find the most occuring arguement patterns.

gSpan is a popular algorithm for discovering frequent subgraphs in a graph database. The input is a set of labeled connected graphs and a threshold named minsup.A labeled graph is a set of vertices and edges, having some labels.The output is the set of all subgraphs that appear in at least minsup percent of the graphs of the input graph database, and their support values.By using this algorithm we would be able to find the most frequent subtrees.

3 Data Analysis

3.1 FP-growth

Following steps were taken into consideration:

- Separate con and pro sentences
- Take 10 percent of the resultant dataset for test
- Get most frequent argument patterns for each using fp-growth algorithm

	support	itemsets
0	0.952381	(56)
1	0.690476	(13)
2	0.666667	(57)
3	0.642857	(34)
4	0.452381	(5)
...	...	...
546	0.261905	(61, 57, 60, 0)
547	0.261905	(56, 61, 60, 0)
548	0.261905	(56, 57, 60, 61)
549	0.261905	(56, 57, 61, 0)
550	0.261905	(60, 56, 57, 61, 0)

551 rows × 2 columns

FIGURE 1: Frequent items from con dataset

- Filter top 10 percent based on the highest support that we got from the previous step
- Analysis to be done on the result (which might return pros or cons depending on the set that we chose in previous steps)

3.2 gSpan

We wanted to classify microtexts by finding the most frequent subgraphs based on their support values.We mined the argument patterns by using gSpan algorithm and got the support values for each of those patterns.One of them with high score value is listed in the figure 2.

Following microtexts *micro_b*021*_con*, *micro_b*051*_con* ,*micro_b*012*_pro*,*micro_k*024*_pro*, *micro_k*027*_pro* are related to the argument pattern mentioned in the figure 2.

```
t # 17
v 0 CC
v 1 _
e 0 1 reb

Support: 43
```

FIGURE 2

3.3 Counting the Pros and Cons of subtrees

One hypothesis was that we could classify based on the number of pro and con relations from all the subtrees present in the microtexts.
It started out rather well. We checked the argument pattern list, and randomly selected long patterns with lots of "pro" relations.
For argument pattern 40 and 63, we all got only "pro" microtexts back, so that was a relief.
Then we checked 65, which seemed like an obvious choice for a pro-like microtext, however, there 2 out of 3 cons.

4 Conclusion

We've come to the conclusion that this task is not very realistic (irregardless of this, we're admitting that we could have done a way better approach to classification) for the following reason:

The microtexts we were supposed to classify had specific argument patterns to them. These argument patterns altogether do not represent the overall "stance", which is either "pro" or "con", rather the relation of the sentences to each other, and finally, to the sentence selected as the main statement against the "topic id".
This main statement, out of the n sentences, will be the one through which we're deciding whether something is "pro" or "con". Take this as an example with argument pattern 65:

Topic Id: "TXL airport remain operational after BER opening", stance = "con".
The main statement was:
"[e6] After the new BER Airport has been commissioned, Tegel should definitely be closed."

All the other sentences were supportive of the main statement, but since the main statement originally went against the topic id, the overall stance became "con".
If we retained the same argument structure, but the topic id was changed to:
"TXL airport does not remain operational after BER opening",
then the stance would have been "pro".

With this in mind, we cannot classify purely based on argument structures, as we need to have the topic id also to see what we're comparing our argument structure to.
If we classified them, like we tried to with FP-growth, we'll just match the structures to labels PRO and CON without knowing what the relation is between the main statement and the topic id.