DDI Baseline

Baseline

General Structure

Resources

Execution and Results

Conclusions

Master in Artificial Intelligence

Advanced Human Language Technologies

UPC

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

FIB

Outline

1 DDI Baseline

2 Baseline

3 General Structure

4 Resources

5 Execution and Results

6 Conclusions

Session 3 - DDI baseline

DDI Baseline
Baseline
General Structure
Resources
Execution and Results
Conclusions

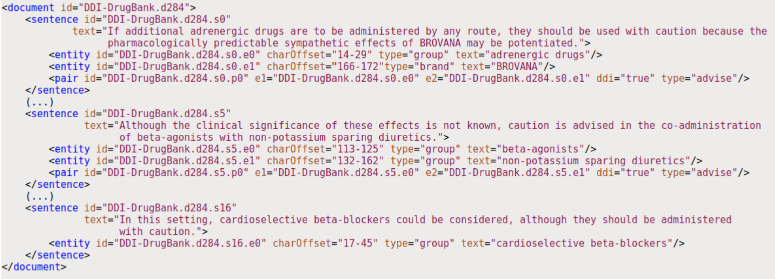
The provided baseline classifies drug-drug interactions between pairs of drugs in the input file. The program uses simple heuristics to carry out the task.

```
$ python3 ./baseline-DDI.py devel.xml devel.out
DDI-DrugBank.d278.s0|DDI-DrugBank.d278.s0.e0|DDI-DrugBank.d278.s0.e1|null
DDI-MedLine.d88.s0|DDI-MedLine.d88.s0.e0|DDI-MedLine.d88.s0.e1|null
DDI-MedLine.d88.s0|DDI-MedLine.d88.s0.e0|DDI-MedLine.d88.s0.e2|null
DDI-MedLine.d88.s0|DDI-MedLine.d88.s0.e1|DDI-MedLine.d88.s0.e2|null
DDI-DrugBank.d398.s0|DDI-DrugBank.d398.s0.e0|DDI-DrugBank.d398.s0.e1|effect
DDI-DrugBank.d398.s0|DDI-DrugBank.d398.s0.e0|DDI-DrugBank.d398.s0.e2|effect
DDI-DrugBank.d398.s0|DDI-DrugBank.d398.s0.e2|DDI-DrugBank.d398.s0.e3|null
DDI-DrugBank.d398.s1|DDI-DrugBank.d398.s1.e0|DDI-DrugBank.d398.s1.e1|null
DDI-DrugBank.d211.s2|DDI-DrugBank.d211.s2.e0|DDI-DrugBank.d211.s2.e5|mechanism
DDI-DrugBank.d211.s2|DDI-DrugBank.d211.s2.e1|DDI-DrugBank.d211.s2.e2|null
...
```

The output must be formatted like this, since it is the format expected by the evaluation script.

Data particularities

Examples



```
<document id="DDI-DrugBank.d284">
    <sentence id="DDI-DrugBank.d284.s0"
            text="If additional adrenergic drugs are to be administered by any route, they should be used with caution because the
                  pharmacologically predictable sympathetic effects of BROVANA may be potentiated.">
        <entity id="DDI-DrugBank.d284.s0.e0" charOffset="14-29" type="group" text="adrenergic drugs"/>
        <entity id="DDI-DrugBank.d284.s0.e1" charOffset="166-172"type="brand" text="BROVANA"/>
        <pair id="DDI-DrugBank.d284.s0.p0" e1="DDI-DrugBank.d284.s0.e0" e2="DDI-DrugBank.d284.s0.e1" ddi="true" type="advise"/>
    </sentence>
    (...)
    <sentence id="DDI-DrugBank.d284.s5"
              text="Although the clinical significance of these effects is not known, caution is advised in the co-administration
                    of beta-agonists with non-potassium sparing diuretics.">
        <entity id="DDI-DrugBank.d284.s5.e0" charOffset="113-125" type="group" text="beta-agonists"/>
        <entity id="DDI-DrugBank.d284.s5.e1" charOffset="132-162" type="group" text="non-potassium sparing diuretics"/>
        <pair id="DDI-DrugBank.d284.s5.p0" e1="DDI-DrugBank.d284.s5.e0" e2="DDI-DrugBank.d284.s5.e1" ddi="true" type="advise"/>
    </sentence>
    (...)
    <sentence id="DDI-DrugBank.d284.s16"
              text="In this setting, cardioselective beta-blockers could be considered, although they should be administered
                    with caution.">
        <entity id="DDI-DrugBank.d284.s16.e0" charOffset="17-45" type="group" text="cardioselective beta-blockers"/>
    </sentence>
</document>
```

Each possible pair of drugs in a sentence is classified as `effect`, `advise`, `mechanism`, `int`, or `null`.

Outline

1. DDI Baseline
2. Baseline
3. General Structure
4. Resources
5. Execution and Results
6. Conclusions

Simple approach

- Extract words in between two entities in target pair.
- For each word, compute conditional probability of each class given word in between.

word	class	$P(c \mid w)$	freq
alcohol	effect	0.52	327
response	effect	0.69	237
interact	int	0.47	198
enhance	effect	0.52	180
action	effect	0.53	130
tylenol	int	0.5	84
due	mechanism	0.62	78
tubular	mechanism	0.42	70
cyp	mechanism	0.81	55
induction	mechanism	0.85	54
cyp2d6	advise	0.475	40
...	...	...	...

- Select words with a high biass for a class (e.g. $P(c \mid w) \geqslant 0.4$) and with a more or less high frequency (e.g. more than 20 occurences).
- Given a new pair, check if the words in between are any of the list, and take a decision accordingly

DDI Baseline

Baseline

General Structure

Resources

Execution and Results

Conclusions

Outline

1. DDI Baseline
2. Baseline
3. General Structure
4. Resources
5. Execution and Results
6. Conclusions

General Structure - Main function

```python
# create tokenizer
nlp = spacy.load("en_core_web_trf",
                 enable=["tokenizer","tagger",
                         "attribute_ruler", "lemmatizer"]))
 tree = parse(datafile)   # parse XML file, obtaining a DOM tree
 # process each sentence in the file
 sentences = tree.getElementsByTagName("sentence")
 for s in sentences :
    sid = s.attributes["id"].value   # get sentence id
    stext = s.attributes["text"].value   # get sentence text
    print(f"processing sentence {sid}        \r", end="")
    tokens = nlp(stext) # tokenize text with spacy lemmatizer
    # load sentence entities
    entities = {}
    ents = s.getElementsByTagName("entity")
    for e in ents :
       id = e.attributes["id"].value
       offs = e.attributes["charOffset"].value.split("-")
       entities[id] = {'start': int(offs[0]),
                       'end': int(offs[-1]),
                       'type': e.attributes["type"].value,
                       'text': e.attributes["text"].value}

    # for each pair in the sentence, decide whether it is DDI and its type
    pairs = s.getElementsByTagName("pair")
    for p in pairs:
       id_e1 = p.attributes["e1"].value
       id_e2 = p.attributes["e2"].value
       ddi_type = check_interaction(tokens, entities, id_e1, id_e2)
       if ddi_type is not None :
          print("|".join([sid, id_e1, id_e2, ddi_type]), file=outf)
```

Outline

1. DDI Baseline
2. Baseline
3. General Structure
4. Resources
5. Execution and Results
6. Conclusions

Resources

The program uses:

- An XML parser: `xml.dom.minidom` (`https://docs.python.org/3.7/library/xml.dom.minidom.html`, included in python standard libray)
- A tokenizer for English text: `SpaCy` (check `https://spacy.io/usage` if you don't have it installed)
- The evaluator module to compute performance scores (provided in the lab project zipfile).

DDI Baseline

Baseline

General Structure

Resources

Execution and Results

Conclusions

Outline

1. DDI Baseline
2. Baseline
3. General Structure
4. Resources
5. Execution and Results
6. Conclusions

Executing the baseline

```
$ python3 run.py
 Running baseline on devel
 Evaluating baseline on devel
 Running baseline on test
 Evaluating baseline on test
```

Results

Results on `devel` dataset

	tp	fp	fn	#pred	#exp	P	R	F1
advise	0	19	143	19	143	0.0%	0.0%	0.0%
effect	27	95	289	122	316	22.1%	8.5%	12.3%
int	29	10	14	39	43	74.4%	67.4%	70.7%
mechanism	9	41	252	50	261	18.0%	3.4%	5.8%
M.avg	-	-	-	-	-	28.6%	19.9%	22.2%
m.avg	65	165	698	230	763	28.3%	8.5%	13.1%
m.avg(no class)	67	163	696	230	763	29.1%	8.8%	13.5%

Results on `test` dataset

	tp	fp	fn	#pred	#exp	P	R	F1
advise	29	50	180	79	209	36.7%	13.9%	20.1%
effect	58	110	235	168	293	34.5%	19.8%	25.2%
int	15	5	25	20	40	75.0%	37.5%	50.0%
mechanism	26	54	318	80	344	32.5%	7.6%	12.3%
M.avg	-	-	-	-	-	44.7%	19.7%	26.9%
m.avg	128	219	758	347	886	36.9%	14.4%	20.8%
m.avg(no class)	141	206	745	347	886	40.6%	15.9%	22.9%

Outline

1 DDI Baseline

2 Baseline

3 General Structure

4 Resources

5 Execution and Results

6 Conclusions

Conclusions

- A simple approach based on words in between the pair entities about 25% average on F1.
- Note that the results are much lower than in the NER task, since DDI is a much more complex task, involving syntax and semantics more than word forms and superficial clues.
- If we use machine learning for this task, we should aim to obtain **significantly** better results, or the additional complexity and cost won't pay off.
- Any ML project requires to start with a **baseline** that sets a threshold to calibrate cost/benefit ratio of the project. The baseline should be a simple knowledge-based or basic statistical approach.
- This will be the goal of the second lab task: **DDI detection**.