University of Antwerp
Faculty of Arts

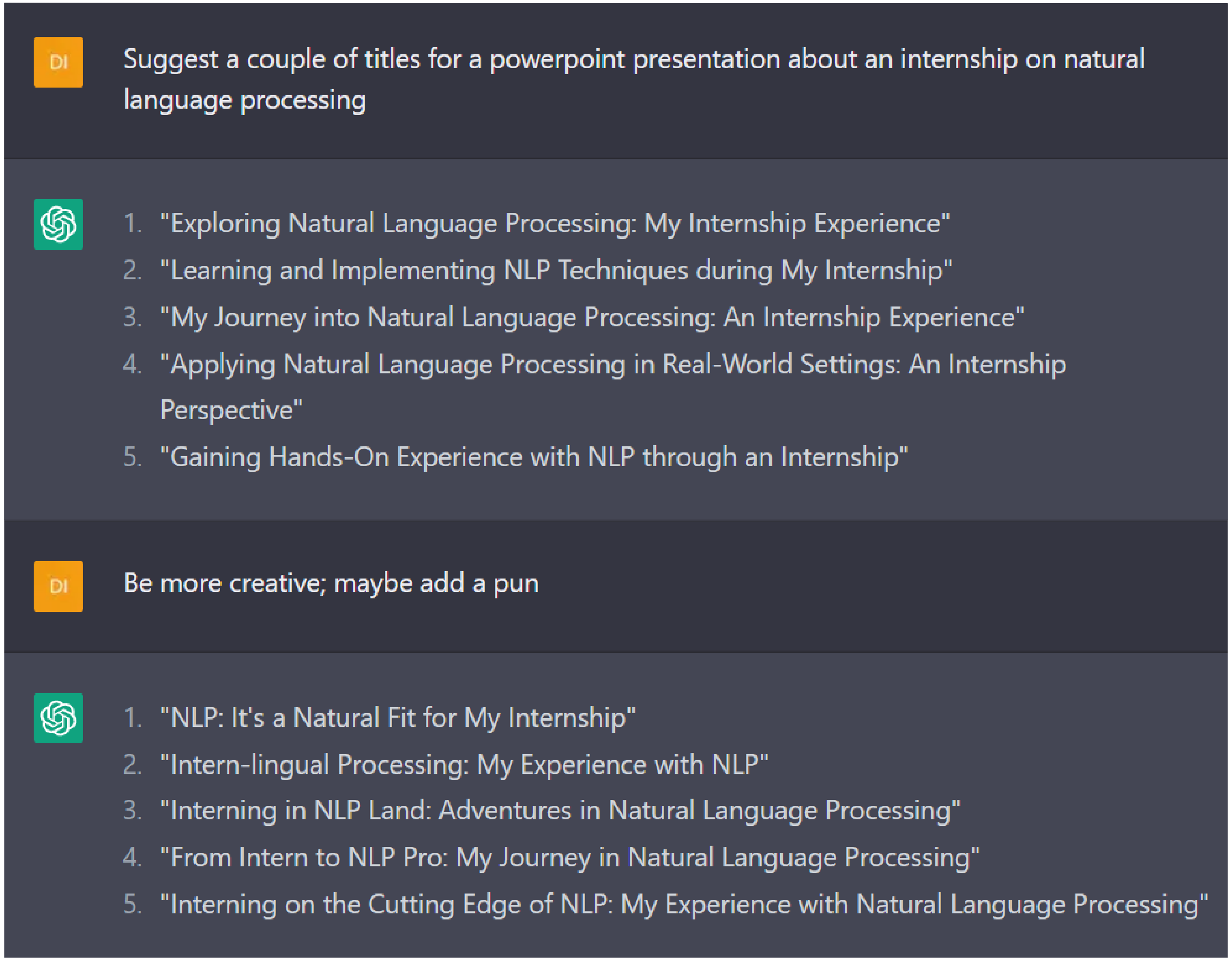
Internship Presentation

Dimitris Boumparis

Supervisor: Prof. Dr. Walter Daelemans

Mentor: Jens Lemmens

08/12/2022



Suggest a couple of titles for a powerpoint presentation about an internship on natural language processing

1. "Exploring Natural Language Processing: My Internship Experience"
2. "Learning and Implementing NLP Techniques during My Internship"
3. "My Journey into Natural Language Processing: An Internship Experience"
4. "Applying Natural Language Processing in Real-World Settings: An Internship Perspective"
5. "Gaining Hands-On Experience with NLP through an Internship"

Be more creative; maybe add a pun

1. "NLP: It's a Natural Fit for My Internship"
2. "Intern-lingual Processing: My Experience with NLP"
3. "Interning in NLP Land: Adventures in Natural Language Processing"
4. "From Intern to NLP Pro: My Journey in Natural Language Processing"
5. "Interning on the Cutting Edge of NLP: My Experience with Natural Language Processing"

General Information

Objective

- Apply the knowledge and skills acquired in the MA DTA in concrete organizational contexts
- **Paths:** research, experiments, programming, application development, etc.

Requirements

- **336 hours** minimum (240 work + 96 self-studying)
- Flexible **conditions** (on-site, WFH)
- combination with thesis

Evaluation

★★☆

- **Portfolio** (2,5k - 5k words)
  - More personal tone
  - Experiences and takeaways
- **Presentation:**
  - More formal tone
  - Results-oriented

Motivation

- Already been an:
  - intern during my bachelor's (translation and localization) (summer of 2017)
  - associate at the Computational Stylistics Lab at NKUA (Prof. Dr. George Mikros) (2017-2019)
- Prior experience in programming (front-end web dev)
- New challenges:
  - Longest stay abroad to date
  - Multi-cultural environment
  - Research centre with international prestige
- Aspirations for a research-oriented career

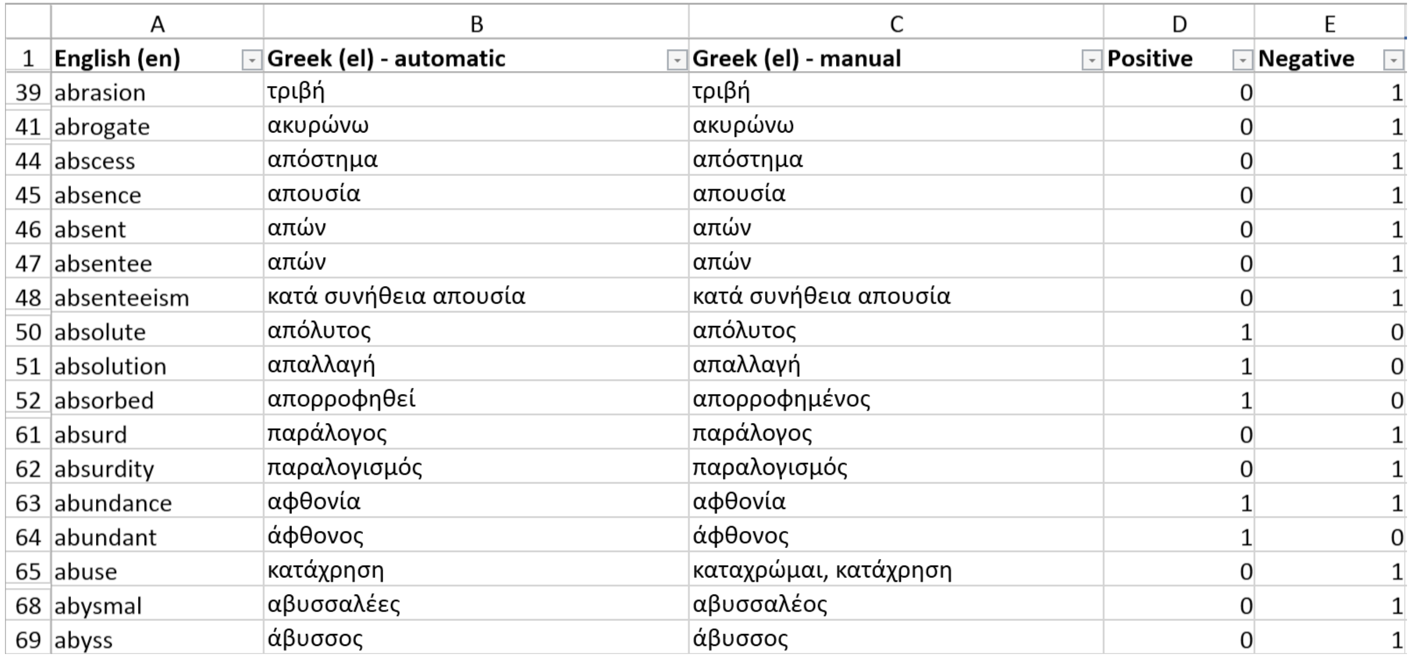
Tasks

- Main: **Author profiling**
- Predict:
  - Gender
  - Age group
  - Personality
  - Job (industry)
- **Classification-only**
- **Algorithms used:**
  - Experimented with traditional machine learning and deep learning (BERT)
  - (Linear) Support Vector Machines was preferred

- Other: **NRC Word-Emotion Association Lexicon** (Mohammad & Turney, 2010)
  - Used in (multilingual) sentiment analysis
  - Version 0.92
    - Translation into Greek
    - MTPE of the Greek MT

NRC Lexicon

- **MTPE** (where column B already populated, ~5,500)

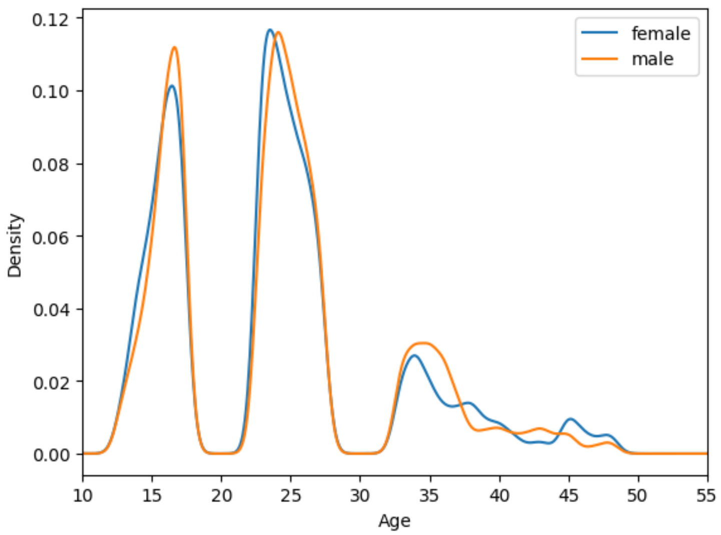


	A	B	C	D	E
1	**English (en)**	**Greek (el) - automatic**	**Greek (el) - manual**	**Positive**	**Negative**
39	abrasion	τριβή	τριβή	0	1
41	abrogate	ακυρώνω	ακυρώνω	0	1
44	abscess	απόστημα	απόστημα	0	1
45	absence	απουσία	απουσία	0	1
46	absent	απών	απών	0	1
47	absentee	απών	απών	0	1
48	absenteeism	κατά συνήθεια απουσία	κατά συνήθεια απουσία	0	1
50	absolute	απόλυτος	απόλυτος	1	0
51	absolution	απαλλαγή	απαλλαγή	1	0
52	absorbed	απορροφηθεί	απορροφημένος	1	0
61	absurd	παράλογος	παράλογος	0	1
62	absurdity	παραλογισμός	παραλογισμός	0	1
63	abundance	αφθονία	αφθονία	1	1
64	abundant	άφθονος	άφθονος	1	0
65	abuse	κατάχρηση	καταχρώμαι, κατάχρηση	0	1
68	abysmal	αβυσσαλέες	αβυσσαλέος	0	1
69	abyss	άβυσσος	άβυσσος	0	1

- Challenges:
  - English is analytical
  - The same form for different purposes (noun == verb) → translations in multiple PoS
  - Polysemous words → only emotional connotations
  - Greek is a cased language → prone to mistranslations

Author Profiling Tasks

- Datasets:
  - **PAN15 Author Profiling Shared Task**
    - Tweets in **English**, Spanish, Italian and Dutch
    - Balanced in genders and ages (see graph)
  - **Blog Authorship Corpus**
    - Short and mid-length tweets/texts (English)

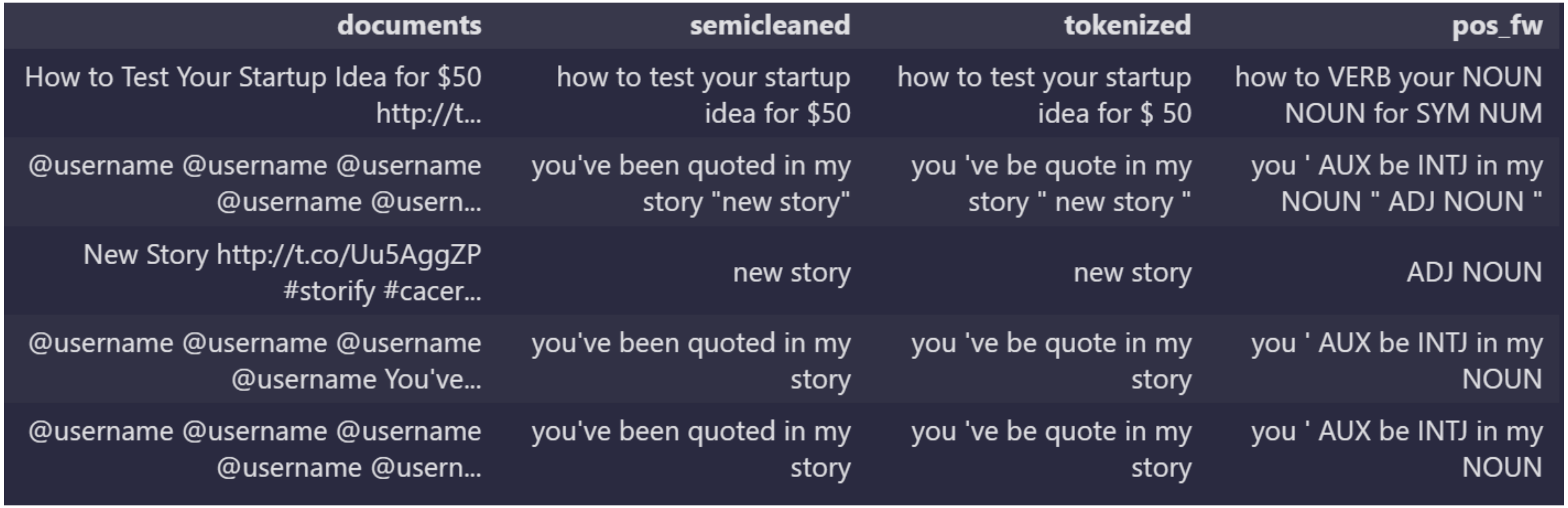


- Compatibility/Sanitization:
  - Age groups
  - Cleaning
  - Tokenization
- **Classifications:**
  - **5 or 10-fold** Grid Search Cross-Validation on F1-macro score
  - Run on CLiPS server or locally
  - Models saved as **joblib** files
  - Used sklearn's **`GroupShuffleSplit()`** for the PAN dataset

Preprocessing

- Lowercasing (stdlib)
- Tokenization (spacy)
- Replace content words with POS tags while maintaining function words

- Different levels of cleaning (RegEx):
  - Remove https://urls, @mentions, #hastags
  - Remove repeated punctuation
  - Replace whitespace with single space

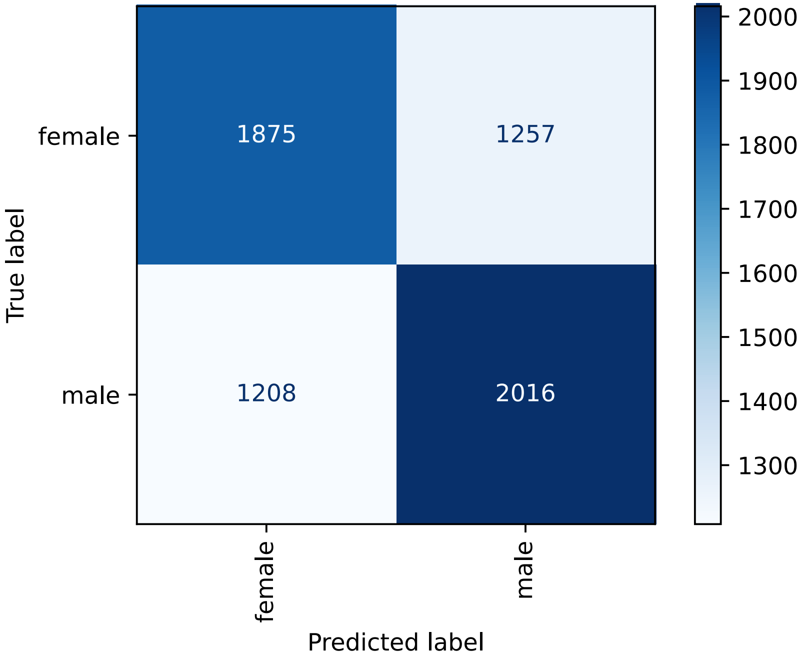


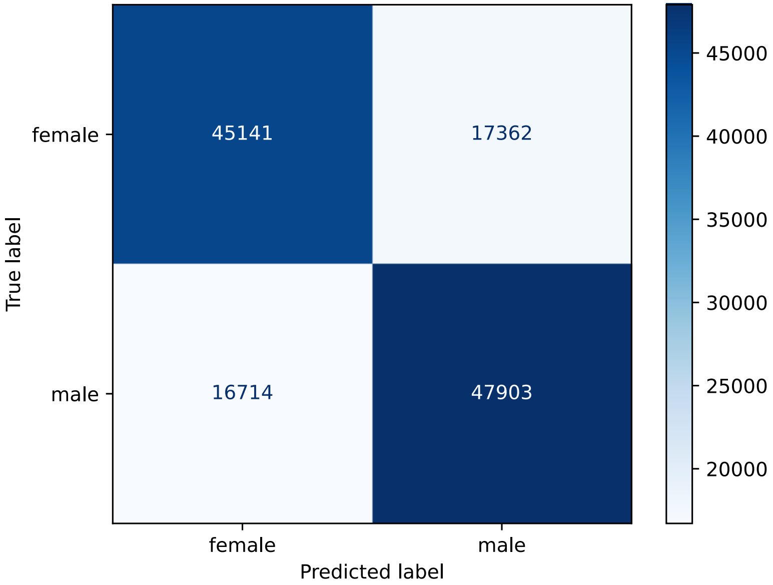
documents	semicleaned	tokenized	pos_fw
How to Test Your Startup Idea for $50 http://t...	how to test your startup idea for $50	how to test your startup idea for $ 50	how to VERB your NOUN NOUN for SYM NUM
@username @username @username @username @usern...	you've been quoted in my story "new story"	you 've be quote in my story " new story "	you ' AUX be INTJ in my NOUN " ADJ NOUN "
New Story http://t.co/Uu5AggZP #storify #cacer...	new story	new story	ADJ NOUN
@username @username @username @username You've...	you've been quoted in my story	you 've be quote in my story	you ' AUX be INTJ in my NOUN
@username @username @username @username @usern...	you've been quoted in my story	you 've be quote in my story	you ' AUX be INTJ in my NOUN

Gender (LSVM / LogReg)

- Binary classification
- LogReg performed significantly worse

- Unigrams: .73
- **1-3grams: .74**
- 1-2grams + POS + Punct: .68
- 1-2grams + POS + Punct + LIWC: .67

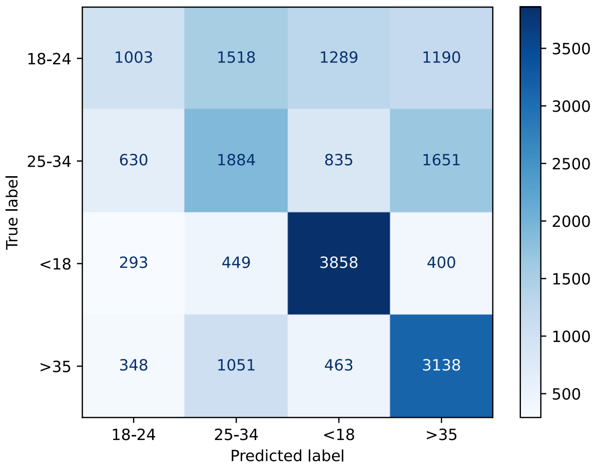


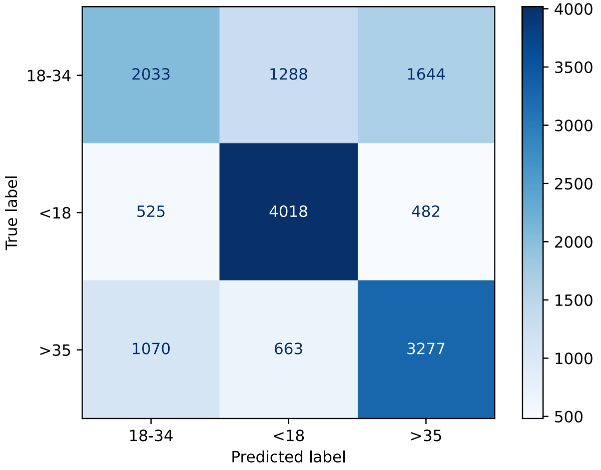


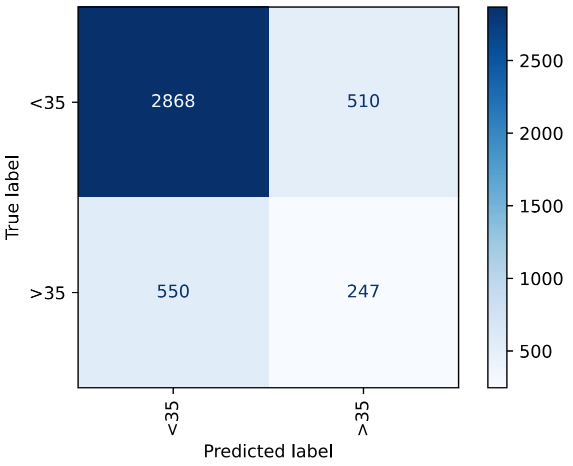
Age group (LSVM)

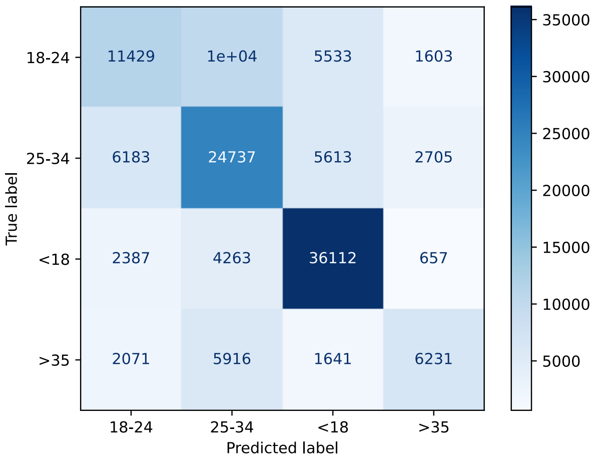
- Groups:
  - Initially: 18-24, 25-34, 35-50, 50+
  - After sanitization: <18, 18-24, 25-34, 35+
  - Finally: <35, 35+ (binary)

- 4 classes: .35, .44, **.52 (1-2grams)**
- 3 classes: .51-**.57 (unigrams)**
- 2 classes: .54 (word/char 1-3grams + FU)
- **4 classes (all samples): .62 (1-3grams)**





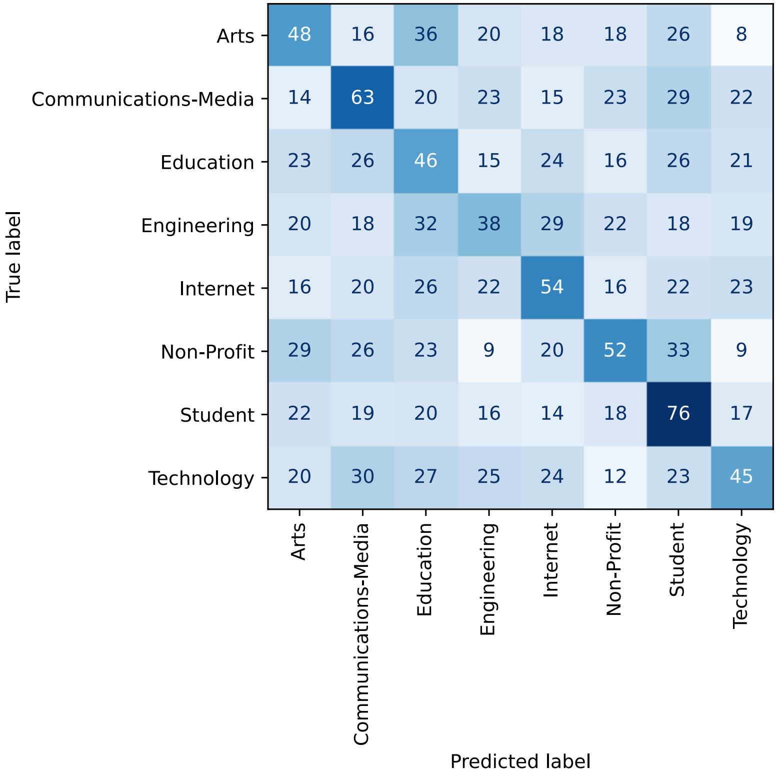




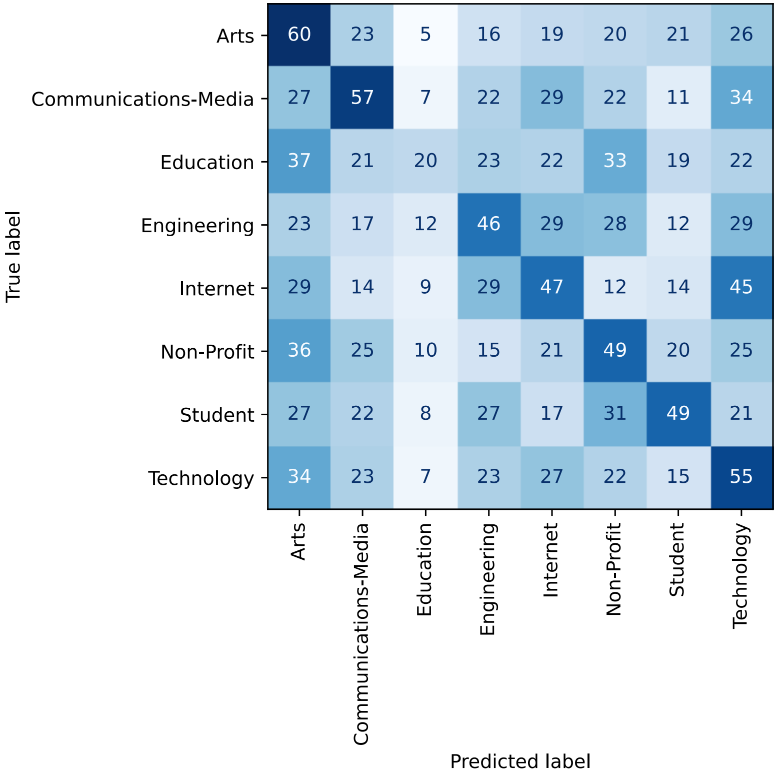
Industry (SVM)

- 36 industries with various # of samples
- 8 with 1,000 samples each

- 1-2grams: .25 (left)
- **All features: .26 (right)**



	Arts	Communications-Media	Education	Engineering	Internet	Non-Profit	Student	Technology
Arts	48	16	36	20	18	18	26	8
Communications-Media	14	63	20	23	15	23	29	22
Education	23	26	46	15	24	16	26	21
Engineering	20	18	32	38	29	22	18	19
Internet	16	20	26	22	54	16	22	23
Non-Profit	29	26	23	9	20	52	33	9
Student	22	19	20	16	14	18	76	17
Technology	20	30	27	25	24	12	23	45

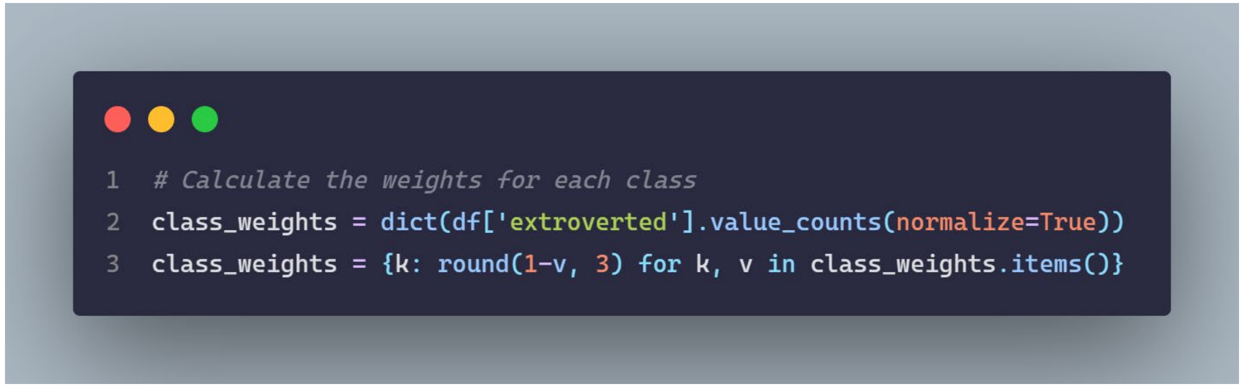


	Arts	Communications-Media	Education	Engineering	Internet	Non-Profit	Student	Technology
Arts	60	23	5	16	19	20	21	26
Communications-Media	27	57	7	22	29	22	11	34
Education	37	21	20	23	22	33	19	22
Engineering	23	17	12	46	29	28	12	29
Internet	29	14	9	29	47	12	14	45
Non-Profit	36	25	10	15	21	49	20	25
Student	27	22	8	27	17	31	49	21
Technology	34	23	7	23	27	22	15	55

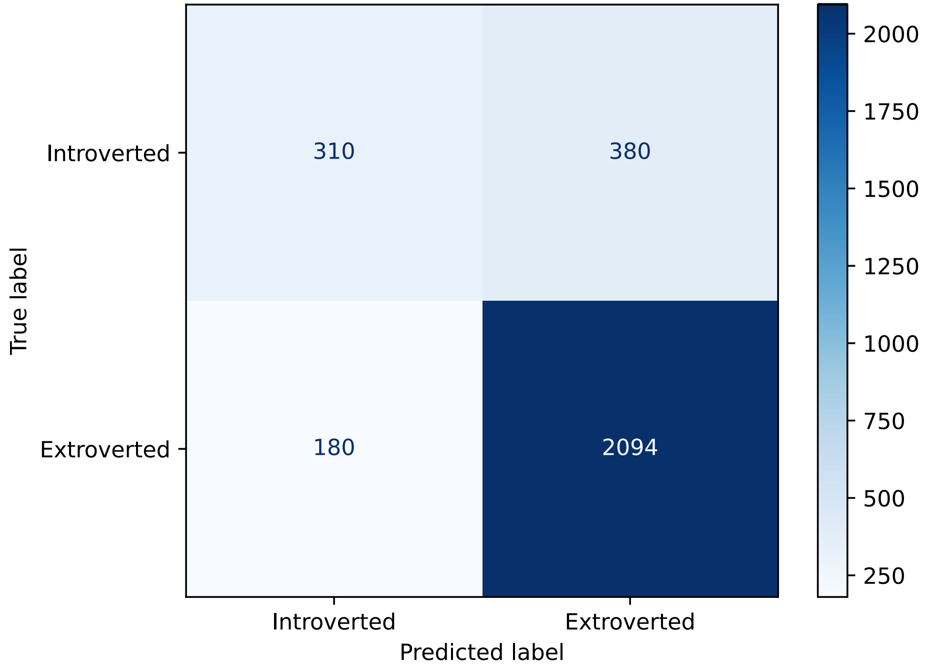
Personality (LSVM)

- Extrovert: -0.5…0.4
- Binary: 1 if score > 0, else 0
- Very imbalanced dataset → adjust weights

- 1-4grams (tokenized text): .56
- **1-3grams (tokenized text) + other personality traits + POS FW: .70**



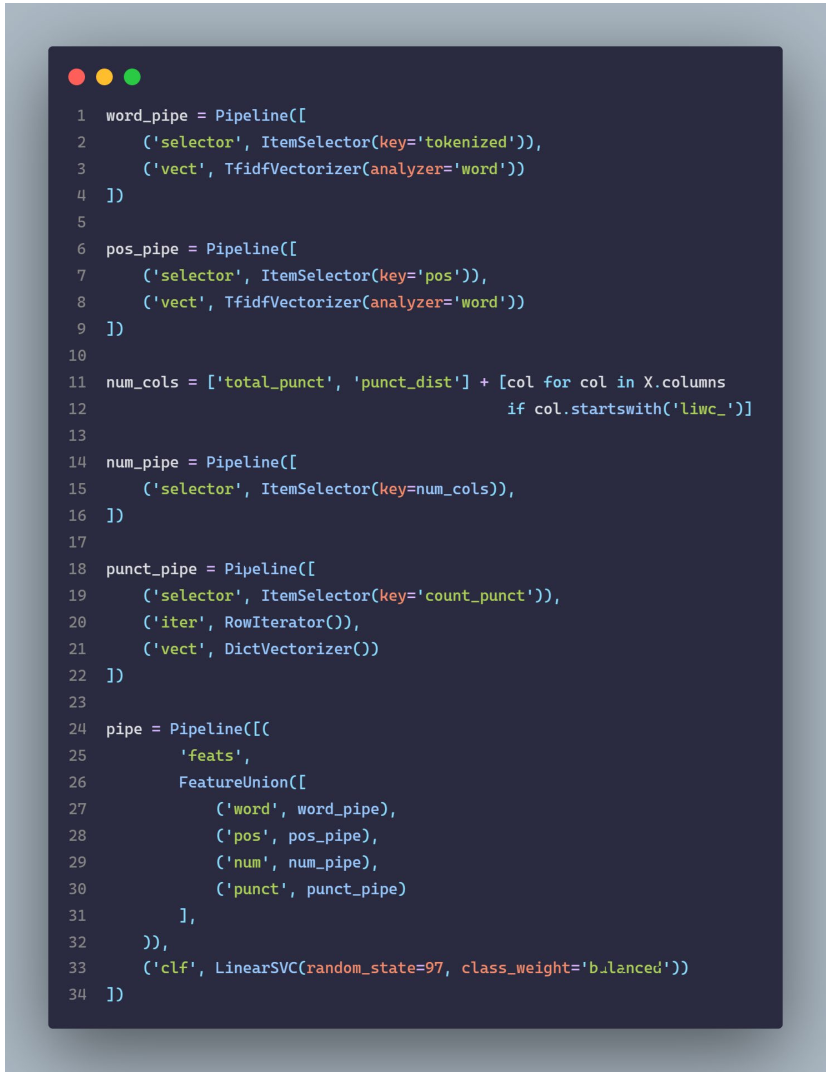
```python
# Calculate the weights for each class
class_weights = dict(df['extroverted'].value_counts(normalize=True))
class_weights = {k: round(1-v, 3) for k, v in class_weights.items()}
```



Scikit-Learn Classes Spotlight

- **FeatureUnion()**
  - Combine several feature extraction mechanisms into a single transformer
    1. TF-IDF word Vectorizer
    2. TF-IDF POS Vectorizer
    3. Punctuation Vectorizer
    4. Numeric Transformers (punctuation and LIWC)

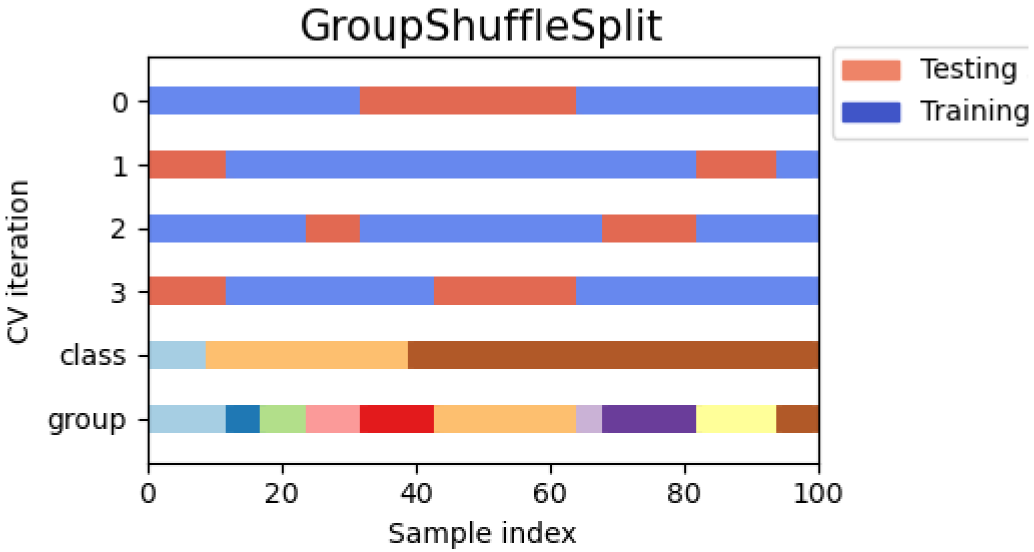
```
from sklearn.pipeline import FeatureUnion
```



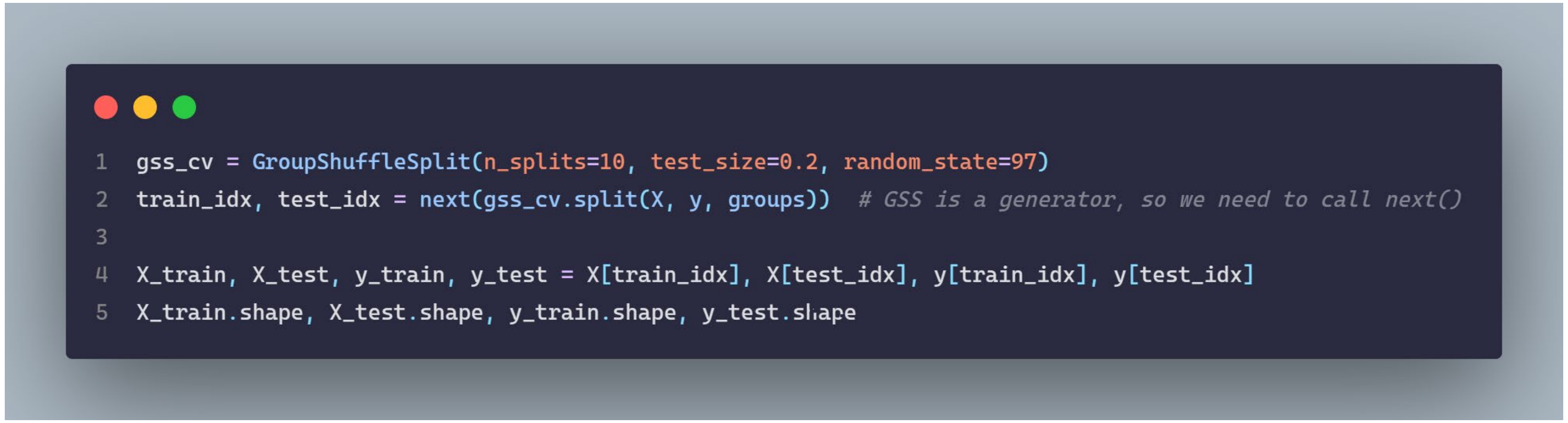
```
word_pipe = Pipeline([
    ('selector', ItemSelector(key='tokenized')),
    ('vect', TfidfVectorizer(analyzer='word'))
])

pos_pipe = Pipeline([
    ('selector', ItemSelector(key='pos')),
    ('vect', TfidfVectorizer(analyzer='word'))
])

num_cols = ['total_punct', 'punct_dist'] + [col for col in X.columns
                                            if col.startswith('liwc_')]

num_pipe = Pipeline([
    ('selector', ItemSelector(key=num_cols)),
])

punct_pipe = Pipeline([
    ('selector', ItemSelector(key='count_punct')),
    ('iter', RowIterator()),
    ('vect', DictVectorizer())
])

pipe = Pipeline([(
        'feats',
        FeatureUnion([
            ('word', word_pipe),
            ('pos', pos_pipe),
            ('num', num_pipe),
            ('punct', punct_pipe)
        ],
    )),
    ('clf', LinearSVC(random_state=97, class_weight='balanced'))
])
```

Scikit-Learn Classes Spotlight (cont’d)

- **`GroupShuffleSplit()`**
  - Shuffle-Group(s)-Out cross-validation iterator
  - Provides randomized train/test indices to split data according to a third-party provided group



```
from sklearn.model_selection import GroupShuffleSplit
```



```
gss_cv = GroupShuffleSplit(n_splits=10, test_size=0.2, random_state=97)
train_idx, test_idx = next(gss_cv.split(X, y, groups))  # GSS is a generator, so we need to call next()

X_train, X_test, y_train, y_test = X[train_idx], X[test_idx], y[train_idx], y[test_idx]
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Conclusions & Takeaways

Tasks

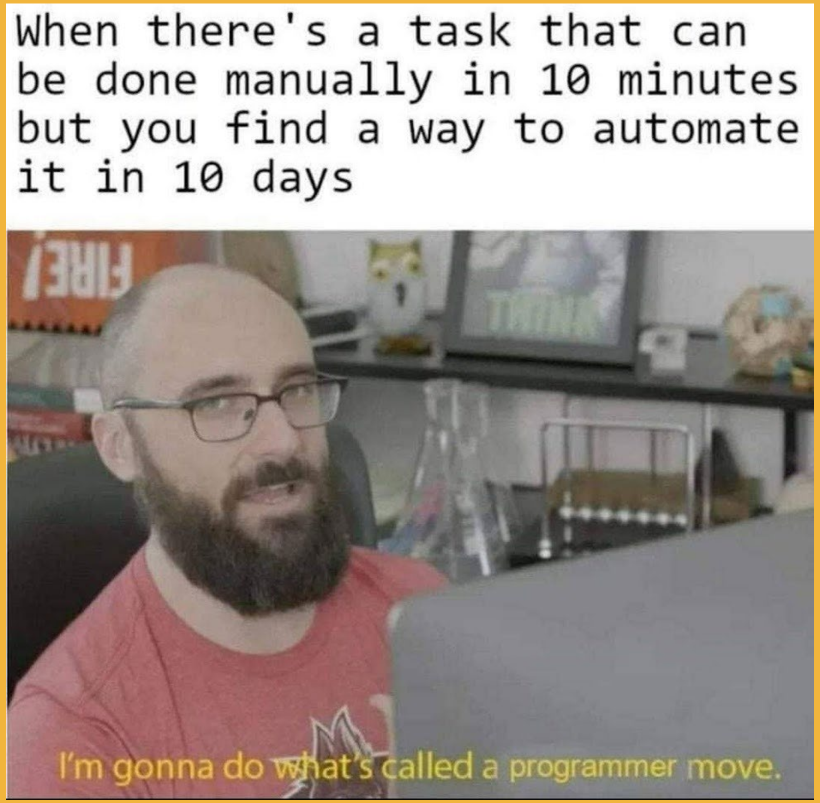
- There's always more to learn in the packages we use every day!
- Tokenization performs better than uncleaned and semicleaned text (kudos to NLP group project)
- Still room to improve models for secondary/tertiary author features

Better data (+ better features) > better models

Overall experience

- Pros:
  - Met awesome people!
  - Participated in very intriguing meetings
  - Tested and expanded the knowledge acquired during the past year
- Cons:
  - Always (or mostly) on-site would be nice
  - Some technical difficulties
  - Put more time on other stuff

Dank u wel!
Nog vragen?



When there's a task that can be done manually in 10 minutes but you find a way to automate it in 10 days

I'm gonna do what's called a programmer move.