We were particularly interested in investigating whether LDA can be used to retrieve a script's internal structure.
Modi et al. also created script templates that described script-specific event labels and participant labels for each scenario (e.g., event labels in BAKING A CAKE: get_ingredients, put_cake_oven, etc. and participant labels: ingredients, oven, etc.), which were used to annotate the stories. They annotated event-denoting verbs in the stories with the event labels and participant-denoting NPs with the participant labels.
The idea was that, assuming a number of topics equal to the number of labels identified in the corpus for a specific scenario, LDA could retrieve a similar structure to the one that was manually annotated in the corpus, namely identify words with the same label as belonging to the same topic.
For this purpose, we tested the model on the 97 stories from the CAKE scenario (37 labels), which seemed to us to display a more uniform, script-like structure across stories. The labels for the CAKE scenario can be manually inspected in: InScript/templates/Cake_scenario_readme.pdf

## Modules

### cake_documents.py

The module was used to:

- extract all the stories from the CAKE scenario in the InScript corpus.
- train the LDA model on 2 versions of the corpus: a regular version, and a version in which, during the 'cleaning' phase (remove punctuation and stopwords, lemmatization) we concatenated the lemmas with the tag they were associated most often with in the InScript annotation. The second version serves visualization purposes: in the pyLDAvis visualization we can more easily recognize whether a topic captures words belonging to the same tag.
  In both cases, we:
- created the term **dictionary** of the courpus, where every unique term is assigned an index.
- converted the list of the documents into Document Term Matrix (**corpus**) using the dictionary prepared above.
- trained the **LDA model** with 37 topics (using the gensim library) on the Document Term Matrix.

The dictionary, the corpus and the lda_model we used are saved to file in the current folder.

The dictionary_with_tags, the corpus_with_tags and the lda_model_with_tags we used are saved to file in the current folder.

# NOTE: LdaModel gives slightly different results every time we run it. Don't run the module if you want to visualize exactly our results.

## cake_documents_eval.py

The module was used to output 'cake_documents_results.txt'. It stores, in order, the 3 tags most often associated with each topic.

## cake_documents_visual.py

This module was meant to show a visualization of the model with pyLDAvis.
While running the model with more than 20 passes, it sometimes returns an error and doesn't display the visualization:
TypeError: (0.012096859242802942+0j) is not JSON serializable
Nevertheless, we preferred running the model with 50 passes and 100 iterations, and refer to the evaluation results.

## cake_documents_with_tags_visual.py

This module was meant to show a visualization of the tagged model with pyLDAvis.
Same as above.

## **Procedure and Scores**

The InScript corpus contains 97 stories instantiating the scenario 'baking a cake'.
We ran the LDA model with 50 passes, 100 iterations and 37 topics on the InScript CAKE corpus (1 document = 1 full story) using the cake_documents.py module.
To evaluate the model, we ran the module cake_documents_eval.py. The results were saved to 'cake_documents_results.txt'.
We here report, the 3 tags most often associated with each topic, ordered from the most frequent:

```
0     ['UnrelEv', 'ScrPart_ingredients', 'ScrPart_utensil']
1     ['UnrelEv', 'ScrPart_ingredients', 'ScrPart_baking_instructions']
2     ['UnrelEv', 'NPart', 'ScrEv_choose_recipe']
3     ['ScrPart_ingredients', 'ScrEv_make_dough', 'Evoking']
4     ['ScrPart_ingredients', 'UnrelEv', 'ScrPart_cake']
5     ['UnrelEv', 'Evoking', 'ScrPart_beneficiary']
6     ['ScrPart_ingredients', 'UnrelEv', 'NPart']
7     ['UnrelEv', 'NPart', 'ScrPart_ingredients']
8     ['UnrelEv', 'ScrPart_beneficiary', 'NPart']
9     ['ScrPart_ingredients', 'UnrelEv', 'ScrPart_cake']
10    ['ScrPart_ingredients', 'ScrEv_make_dough', 'ScrPart_utensil']
11    ['ScrPart_ingredients', 'ScrPart_cook', 'ScrPart_utensil']
12    ['UnrelEv', 'ScrPart_ingredients', 'ScrEv_get_ingredients']
13    ['ScrPart_ingredients', 'UnrelEv', 'Evoking']
14    ['ScrPart_ingredients', 'ScrPart_utensil', 'UnrelEv']
15    ['ScrPart_ingredients', 'UnrelEv', 'NPart']
16    ['UnrelEv', 'ScrPart_ingredients', 'ScrPart_utensil']
17    ['UnrelEv', 'ScrPart_ingredients', 'ScrPart_baking_instructions']
18    ['UnrelEv', 'ScrPart_ingredients', 'NPart']
19    ['ScrPart_ingredients', 'UnrelEv', 'ScrPart_baking_instructions']
20    ['ScrPart_ingredients', 'UnrelEv', 'ScrEv_make_dough']
21    ['UnrelEv', 'ScrPart_cook', 'ScrPart_ingredients']
22    ['ScrPart_utensil', 'ScrPart_ingredients', 'ScrPart_cake_tin']
23    ['ScrPart_ingredients', 'ScrPart_cake', 'UnrelEv']
24    ['UnrelEv', 'ScrPart_ingredients', 'NPart']
25    ['UnrelEv', 'ScrPart_ingredients', 'NPart']
26    ['UnrelEv', 'ScrPart_ingredients', 'NPart']
27    ['UnrelEv', 'NPart', 'ScrPart_ingredients']
28    ['UnrelEv', 'ScrPart_ingredients', 'ScrEv_make_dough']
29    ['UnrelEv', 'ScrPart_ingredients', 'ScrPart_cake']
30    ['ScrPart_ingredients', 'UnrelEv', 'ScrPart_utensil']
31    ['ScrPart_ingredients', 'UnrelEv', 'ScrPart_cake']
32    ['UnrelEv', 'ScrPart_ingredients', 'ScrPart_baking_instructions']
33    ['ScrPart_ingredients', 'UnrelEv', 'ScrPart_utensil']
34    ['ScrPart_ingredients', 'UnrelEv', 'ScrPart_utensil']
35    ['NPart', 'ScrPart_kitchen/location', 'ScrPart_ingredients']
36    ['ScrPart_ingredients', 'UnrelEv', 'ScrEv_make_dough']
```

We here report the distribution of the first best, the second best and third best tags across positions:

first best tags
'NPart': 1
'ScrPart_ingredients': 17
'ScrPart_utensil': 1
'UnrelEv': 18

second best tags
'NPart': 3
'ScrPart_cake': 1
'Evoking': 1
'ScrPart_ingredients': 13
'ScrPart_beneficiary': 1
'ScrPart_kitchen/location': 1
'ScrPart_cook': 2
'ScrEv_make_dough': 2
'ScrPart_utensil': 1
'UnrelEv': 12

third best tags
'NPart': 7
'ScrPart_ingredients': 4
'ScrEv_choose_recipe': 1
'ScrPart_baking_instructions': 4
'ScrEv_get_ingredients': 1
'ScrPart_beneficiary': 1
'ScrPart_utensil': 7
'ScrPart_cake': 4
'Evoking': 2
'ScrEv_make_dough': 3
'ScrPart_cake_tin': 1
'UnrelEv': 2

## Results

We were looking at whether 37 topics as retrieved by the LDA model could overlap with the 37 tags used to annotate participants and events in the InScript CAKE corpus.
18 out of the 37 first best tags associated with each topic are irrelevant to the cake scenario ('UnrelEv' stands for Unrelated Event, and, together with 'Npart', is not part of the 37 labels specific to the cake scenario). Another 17 are 'ScrPart_ingredients' (one of the participants labels). One is 'ScrPart_utensil'. This suggests that the first best tags are rather uninformative when it comes to unveiling the internal script structure. In fact, they rather reflect characteristics pertaining to the scenario as a whole.
The results are analogous for the second and third best tags.
The second best tags cover only 7 out of 37 tags relevant to the scenario ('ScrPart_cake', 'ScrPart_ingredients', 'ScrPart_beneficiary', 'ScrPart_kitchen/location', 'ScrPart_cook', 'ScrEv_make_dough', 'ScrPart_utensil').
The third best tags cover only 9 ('ScrPart_ingredients', 'ScrEv_choose_recipe', 'ScrPart_baking_instructions', 'ScrEv_get_ingredients', 'ScrPart_beneficiary', 'ScrPart_utensil', 'ScrPart_cake', 'ScrEv_make_dough', 'ScrPart_cake_tin').
We didn't find the overlap we were looking for. On the other side, the fact that the number of tags covered increases at lower levels, and that more specific event tags start to appear ('ScrEv_make_dough' as opposed to 'ScrPart_ingredients'), seems to suggest that, if any internal structure can be revealed, it might be intercepted at lower levels.