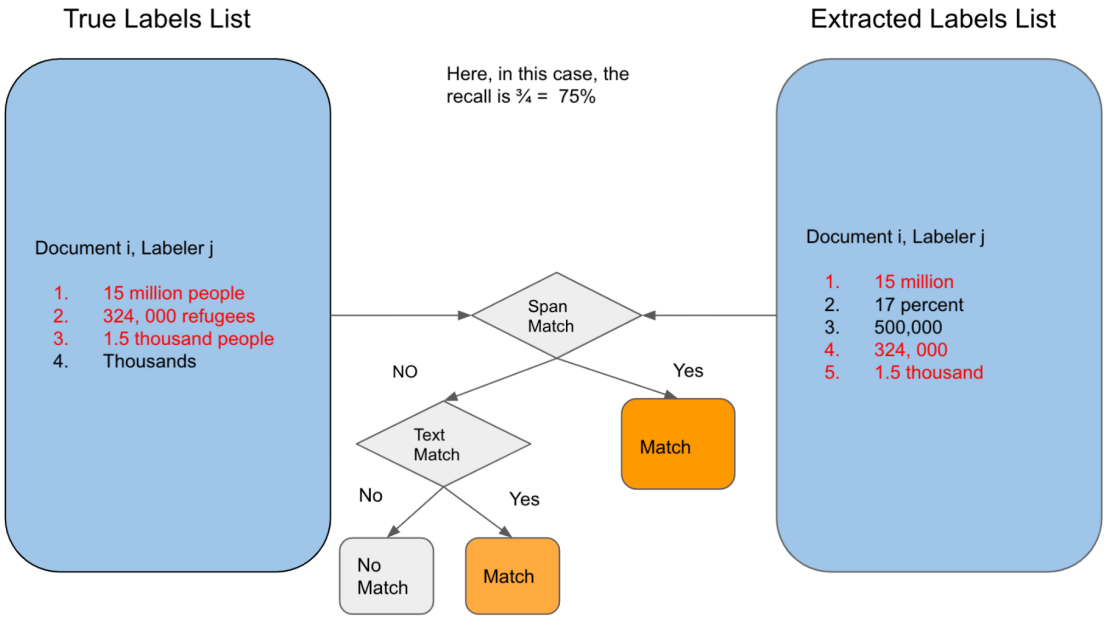
Develop the methodology to analyse the performance of the NLP, especially on location, date and quantity

Introduction:

The goal of this task is to extract information of displacement, e.g. location, date, and quantity. That is, given an article, we can extract displacement related information involving location, date, and quantity and mark the corresponding position in the article.

The benchmark model developed by IDMC was based on the dependency parsing tree model, which was constructed by spaCy.

Evaluation Metric:



The evaluation metric that we use is recall. That is if there is any overlap between predicted text span and the true span or if the predicted content string is the substring of ground truth content or if the ground truth content is substring of the predicted content string.

Here is an example of how our evaluation metric processes. The left hand side is the true label list and the right hand side is the extracted (predicted) label list. Given document i and labeler j, if there exists an interception between the span of the true label and that of the extracted label, it would just return matched. Otherwise, it would further check whether there is a match in content. For example, the text content “15 million” in the extracted label list is the substring of

“15 million people” in true label list, and so they are matched. Therefore out of 4 cases in total, there are 3 cases in the true label list that found their matches. Here the call is ¾ = 75%

Location:

For extracting related location from an article, the basic idea of IDMC’s approach is that first split the whole sentence into different entities and then check whether the corresponding label is about location. We followed their basic idea but did some improvements in the data preprocessing part, which involves unifying and lemmatizing words and expanding the location corpus. As a result, we can detect some corner case locations, and recall enhanced from 50.88% to 71.12%. On the other hand, right now both the benchmark algorithm and ours cannot detect the location if the label of the location entity is not in “GPE”. (Add an example of fail case)

Here is an example of a case where both the benchmark algorithm and ours fail to work.

Original text:
\n\nIn California, more than 120 visitors and staff members were rescued Thursday after being trapped by up to 7 feet (2 meters) of snow in a Sierra Nevada resort for five days.

In this case, the true location label is “Sierra Nevada”. The algorithm fails because this location is too specific to be incorporated into the “GPE” location corpus. Therefore, the algorithm cannot detect this location.

Quantity:

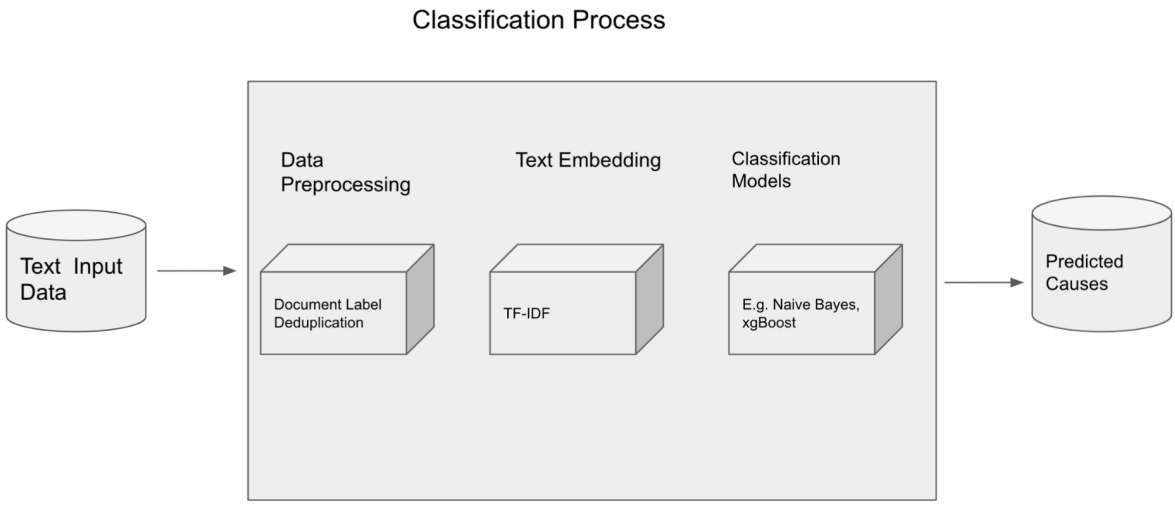
For extracting related quantitative words from an article, the basic idea is to break the whole sentence into several noun phrases, and then iterate each token to see if the given phrase contains quantitative words. For our improvement, we can detect several consecutive quantitative words given a noun phrase instead of only detecting a single word. For example, for the noun phrase, “9.5 million people”, we know the answer would be “9.5 million”. However, in their algorithm, as long as it detects a quantitative word, which is “9.5”, it will stop. On the other hand, for our approach we will search until the next word is no quantitative word then stop, so for this scenario, we will return “9.5 million”. Also, in our algorithm, we incorporated some text cleaning process, which involved removing unnecessary punctuation, e.g. “\n\n”, and lemmatizing words. As a result, the recall enhanced from 69.69% to 75.38%

Date:
For extracting related date from an article, we did not do much change and just kept the benchmark algorithm from IDMC. We constructed the evaluation process for date information extraction and got recall 87.5%.

Cause Classification:

Goal:
The goal of giving an article, we want to know the reason for displacement, whether it comes from conflict or disasters.

Method:



Since there are only 2 out of 83 documents whose labels are marked as “OTHERS” instead of conflict or disasters, and also the category “OTHERS” by definition, means it is ambiguous to be defined as either “CONFLICT” or “DISASTER”, we directly remove the documents that are labeled as “OTHERS” and treat this classification problem as binary.

After the data preprocessing stage involving document label deduplication and text cleaning, we used the TF-IDF method for text embedding and models involving Naive Bayes, xgBoost for classification. For the evaluation process, we used “AUC” score as our evaluation metric and leave-one-out method for cross validation.

Here are the evaluation results. This is the performance table among different classification models in validation set and test set.

	Validation	Test
Multinomial Naive Bayes	0.967	0.976
xgBoost	0.883	0.907
Random Forest	0.939	0.951
Logistic Regression	0.964	0.972
Support Vector Machine	0.964	0.973

Type Multi-Language Classification:

The goal of this task is similar to the Displacement Cause Classification. That is given a document, our goal is to predict whether it is relevant to topic displacement, but in this case, we have different language versions of articles, which involves English, French, and Spanish.

We have three approaches to this relevance classification problem.

- Approach 1: Native-Langauge models
    - We treat each language separately. That is we train English document model, French document model and Spanish model.
    - For this approach, we just feed the embedding text input into our classification pipeline.

- Approach 2: Merge French and Spanish text document into English one
    - Since the text embedding model that we used is TF-IDF, which is a word frequency based algorithm, no matter the english word or french word, they can be treated as a text feature. Therefore, we can just merge multi-languages text sources together and train a whole model for relevance classification.

- Approach 3: Translate French and Spanish into English, merge text documents together and train a whole model.
    - We apply mbart50_m2m model from Facebook research for machine translation
    - Used the same pipeline and evaluation method as we did for cause classification.

Here are the results for the three approaches.

Approach 1

Native-Langauge models

Spanish:

	Validation	Test
Multinomial Naive Bayes	0.865	0.876
xgBoost	0.843	0.853
Random Forest	0.870	0.881
Logistic Regression	0.880	0.890
Support Vector Machine	0.870	0.121

French:

	Validation	Test
Multinomial Naive Bayes	0.832	0.863
xgBoost	0.719	0.765
Random Forest	0.803	0.850
Logistic Regression	0.850	0.880
Support Vector Machine	0.841	0.127

Approach 2:

Merge French and Spanish text document into English one

	Validation	Test
Multinomial Naive Bayes	0.789	0.783
xgBoost	0.738	0.756
Random Forest	0.807	0.814
Logistic Regression	0.850	0.851
Support Vector Machine	0.770	0.238

Approach 3:

Translate French and Spanish into English:

	Validation	Test
Multinomial Naive Bayes	0.764	0.762
xgBoost	0.830	0.831
Random Forest	0.831	0.834
Logistic Regression	0.846	0.848
Support Vector Machine	0.747	0.262

Conclusion:

- Among all the type classification approaches, the logistic regression model is the best. By comparing each approach, the first approach which trains the model of different languages separately works the best, and it reaches 89% AUC score for Spanish and 88% AUC score for French.

- By Comparing the performance of approach 2 and approach 3, the translation process improved the result of some models involving xgBoost and Random Forest. However, it does not help improve the best model logistic regression.