Electronics and Computer Science
Faculty of Physical Sciences and Engineering
University of Southampton

COMP3222 Coursework

MediaEval Tweet Classifier

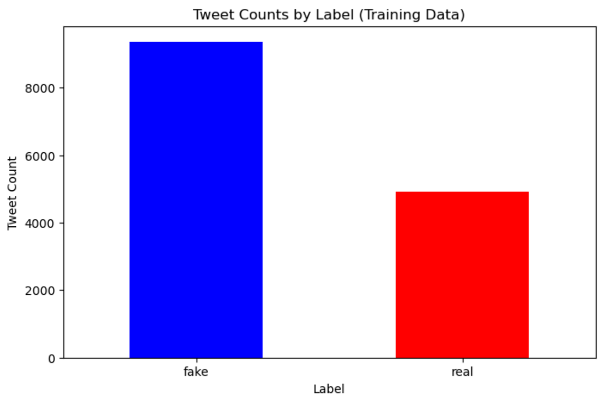
Daniel-Dragos Braghis (32161204)
12 January 2024

Introduction and Data Analysis

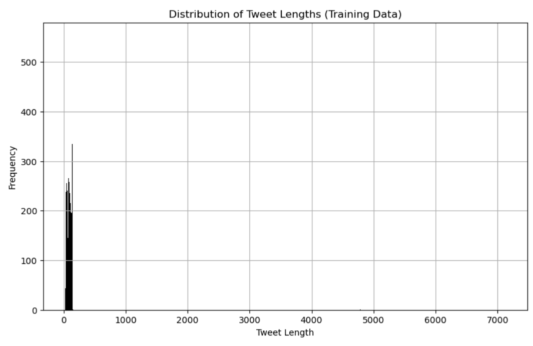
The MediaEval 2015 "verifying multimedia use" task is a classification problem aiming to automate the detection of real vs fake posts on social media platforms like Twitter. Raw training and testing datasets are supplied in text format. While preprocessing is allowed, it is important to note that all the fields in the training set had to be used for the model evaluation. Humour labels are considered fake posts in this scenario. The F1 score is specified as the success metric of the classifier.

On the first inspection, the following tab-delimited columns are available: tweetId, tweetText, userId, imageId(s), username, timestamp and label. Notably, the tweetText uses various languages with symbols, emojis, hashtags and user references that will be valuable information in deciding how to categorize the text. Moreover, the fake and real labels hint at the use of supervised learning techniques as an efficient way to approach the problem.

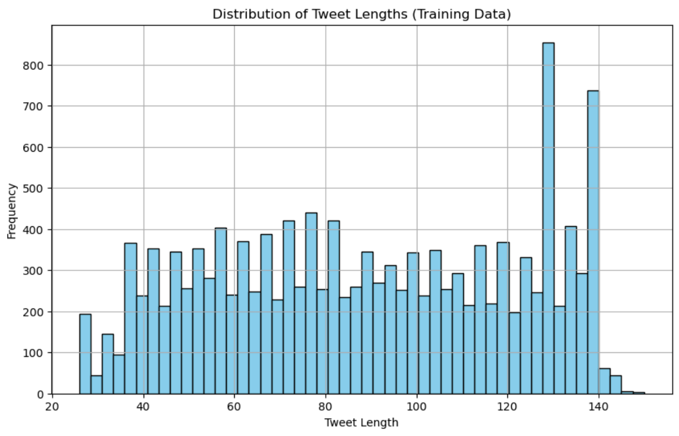
Once importing the data into tables, general information about the dataset is available (the training set will be implicitly referenced throughout the report). Regarding data quality, additional observations can be made: tweets end in a URL, text frequently contains typos and there is no missing data. The training set provided contains 14277 entries while the training one 3755 entries. When counting real and fake entries there are 65.5% fake entries in the training set and duplicate tweet texts predominantly in the training set - 1901 vs 49 in testing, which is a bias to consider when training the model. The same bias was observed in the testing data.



More in-depth analysis followed with tweet-length analysis by plotting the tweet length distribution:

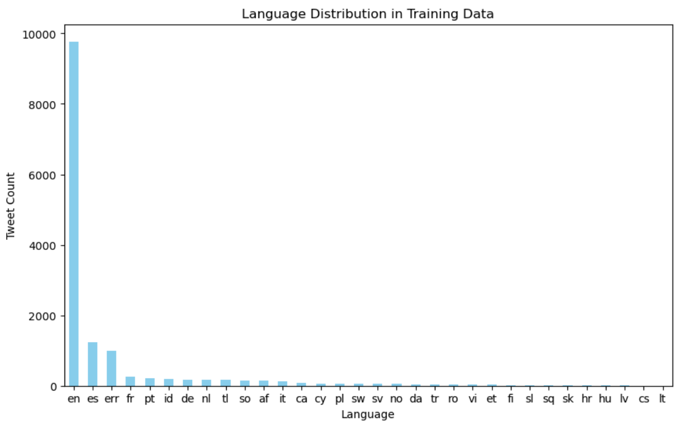


This revealed that the majority of data has a length well below 500 with few outliers with a much higher length. The vast majority of tweets are below 150 words as shown:



Displaying one of the outliers revealed that the long character tweets are incorrectly read entries from the original file (probably not using tabs as delimiter). Such instances will need to be removed in the final training set.

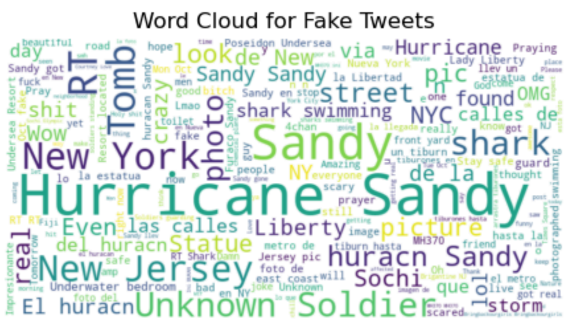
Language analysis of the cleaned text (only text) using the langdetect Python library indicates a heavy presence of English as expected but many other languages are present in the dataset:

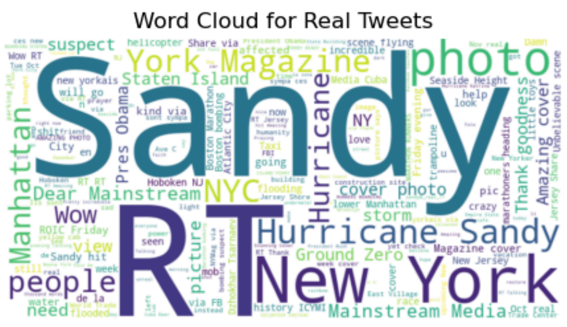


Other languages still constitute more than one-third of the entries in the training set so translation into English would be a preferred preprocessing step before training. Language detection also assigned confidence scores for the language detected between 0 and 1. Based on data distributed into bins (Appendix 1), a confidence level above 0.7 offers reasonable accuracy in detecting the language.

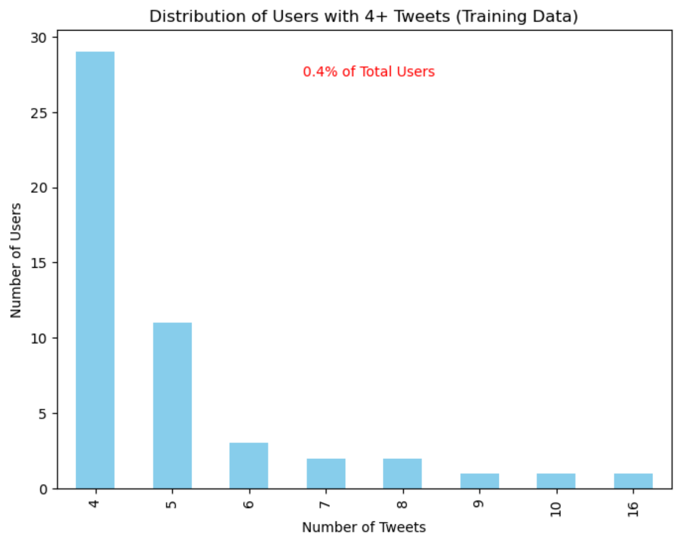
Timestamp analysis, surprisingly, has not revealed useful trends when posting real vs fake tweets as both followed similar behaviours. Not a significant feature to consider.

Text content however varies significantly with real tweets putting a lot more emphasis on a small set of words (Sandy, Hurricane, New York) compared to fake tweets which have a more even distribution of words in their texts. RT (retweet) is a significant feature to consider in this context as fake tweets are rarely retweets of other users or bots as they are “original” fabrications.

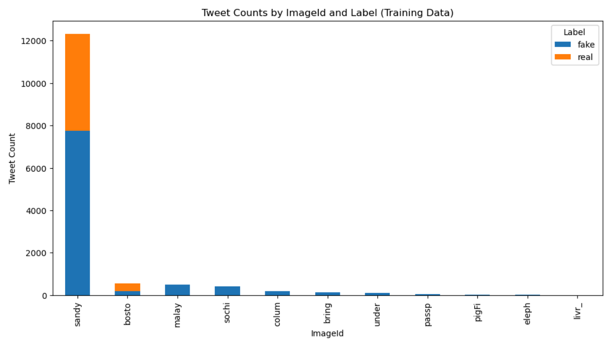


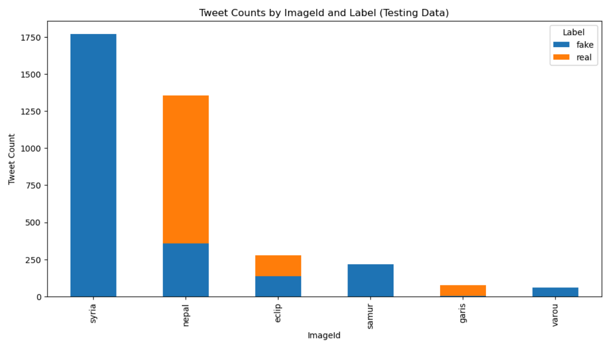


An analysis of userId to identify users that tweet a lot as spammers shows that only 0.4% of total users tweeted more than 4 times making it an insignificant feature for our purposes.



Plotting tweet counts by imageId stressed the importance of the model to generalize well as the training and testing data have very different contexts for the tweets as they are attributed to different events - training dominated largely by Hurricane Sandy and test not even containing it.





Pipeline Design

Following the in-depth exploratory phase, a general pre-processing pipeline was designed. Malformed tweets with a length greater than 150 characters are removed. Cases in which the URL is using escape characters are removed. Finally, following language detection, where the language failed to be detected (marked with ‘err’ text) and cases with low confidence (less than 0.7) are also removed from the training data. Testing data has no such filtering applied as it is used entirely for final assessment.

Spell checking was attempted to improve text translation but because of its resource intensiveness, this preprocessing step is skipped. Translation using the deep_translator library is then applied to the original and clean texts as both may prove useful as features for different classifiers and to avoid redundancy. Only one textual feature is used for training the classifier in combination with other numeric features. As translation is a resource-intensive task, the results post-translation are saved to CSV files for training and testing.

These numeric features are counts for hashtags, emojis, mentions, exclamation marks, and question marks [6]. As previously noted, a binary feature (0 or 1) denoting that the text is a retweet is added.

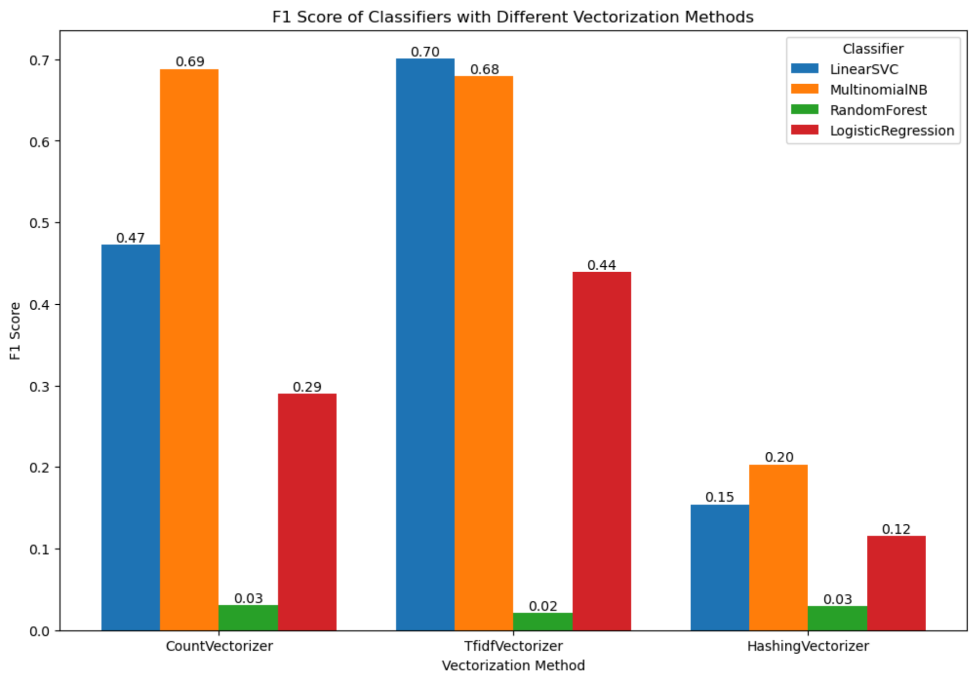
Sentiment analysis is then applied to the clean text using the textblob library with polarity from 0 (negative) to 2 (positive) as some classification algorithms (e.g. Naive Bayes) take as input only positive numeric values and subjectivity from 0 (objective) to 1 (subjective).

Feature normalisation was considered for numeric data but most of the values are already in single up to double digits (outliers) for all symbols so this step was considered unnecessary and not expected to significantly influence the model performance.

Finally, the original tweetId, userId, username and imageId(s) features are removed as they uniquely identify an entry and don’t provide any useful information to the model. In some instances, it may even cause overfitting and poor generalisation of the model as, for example, the imageId(s) for Hurricane Sandy in the training set are not present in the test set at all. Moreover, the timestamp was processed and split into year, month, day and hour (analysis on it hasn’t proved any fake vs real trends) and the original timestamp column was removed. Other meta columns used for preprocessing (language, confidenceBin, contiansProperUrl) are also not included in the training set.

Note that, coincidentally, after preprocessing the bias between the fake and real entry count is still at 65%.

We opted to build a general pipeline including multiple vectorizers and classifiers trained only on the original tweet text to decide on the final two most promising classifier pipelines to optimize. Following research from the wider literature, Linear SVC (SVM-based algorithm) [3,4,6], Logistic Regression [7], Random Forest [1,2,4,6] and Naive Bayes [4] are classic supervised learning algorithms that were successfully applied to this data corpus in the past. Count (Simple Bag Of Words) [5] and TF-IDF (Prioriting unique words) [8] vectorizers are popular choices we test but also experimentally included the hashing vectorizer [8] for comparison. Running the pipeline with basic settings and weighted classes for our biased data (towards fake):



As seen in the figure, the best performance without any optimizations was achieved with SVM and Naive Bayes, with Linear Regression barely being feasible "out of the box". Random Forest needs a lot of optimization and a very different (more complex) pipeline to work with our data. Also, the Hashing Vectorizer performed much worse probably because of the loss of information in the text - not a good fit for our case unless used in conjunction with other features.

The LinearSVC and Naive Bayes will be the two models used for further improvement.

Evaluation

Before optimizing further, a validation set composed of 20% of the training entries was split to be used during model evaluation. This was done to spot data overfitting and guide optimization even if the dataset is relatively small and this could hurt model training.

Random seed used for validation splitting: 42 🙂

1. LinearSVC

Initial performance for Count and TF-IDF respectively (other comparisons will follow the same pattern):

Validation F1 Score: 0.903
Test F1 Score: 0.862

Validation F1 Score: 0.920
Test F1 Score: 0.854

First, it was observed that increasing the max_features parameter (the maximum amount of words to consider) and including English stop words in the vectorizers affected significantly the classifier performance. Incrementing the max feature by 1000 until the performance plateaus or decreases and removing the cap on TF-IDF led to the following improvement:

Validation F1 Score: 0.903
Test F1 Score: 0.862
Validation F1 Score: 0.921
Test F1 Score: 0.787

Next, we tried the other textual features engineered in the preprocessing stage beside the original tweet text. With no additional features, cleaned and/or translated text marginally decreases the performance of the classifier. Translating and cleaning the text might be removing contextual information.

Also, including other features for the classifier training hurt accuracy (because of the timestamp features that are not scaled to the other features). Trying to scale the feature has not helped so there might be high collinearity between features, even when using translated and/or cleaned text in conjunction with various combinations of features the model does not improve in performance.

Trying to optimize the C, gamma, or loss function of the LinearSVC has not led to better F1 scores either. Coincidentally, mostly default values proved to be a good configuration for the training data resulting in the highest F1 Score when testing of 0.862.

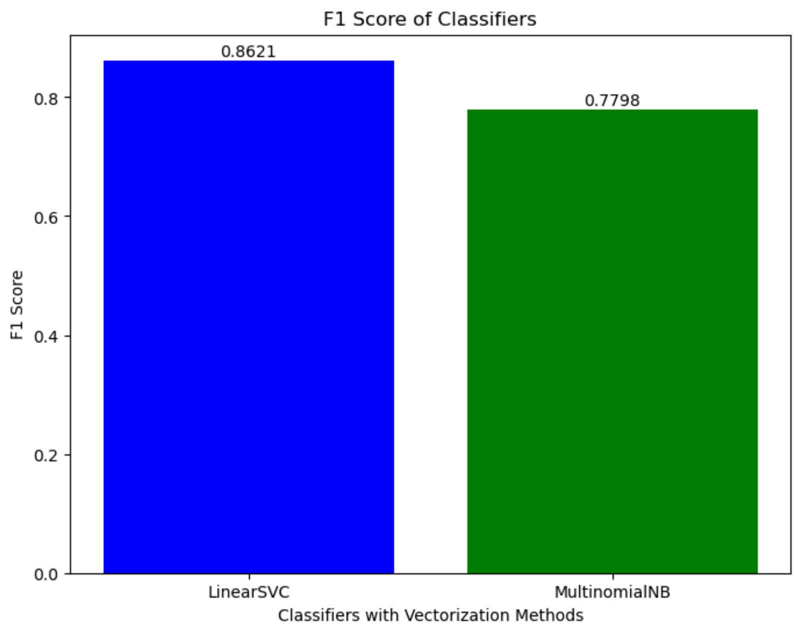
2. Naive Bayes

Naive Bayes has not delivered an improvement over LinearSVC after following a similar optimization process with the TF-IDF vectorizer on the original tweet text.

Validation F1 Score: 0.825
Test F1 Score: 0.779

Conclusion

Here are the final results of our two classifiers:



Both classifiers perform better than a dumb algorithm (guessing) with the LinearSVC achieving the highest F1 score with original tweet text as input for the model. Curiously, adding dimensionality to the problem has not improved the scores for both classifiers. Optimizing the vectorization method was to a large degree enough to achieve the results above.

Text cleaning and translation did not affect the quality of the data after vectorization in a positive way. Keeping symbols, especially emojis, was valuable information best expressed during the vectorization of the text rather than a separate numeric feature [4]. Perhaps, normalizing and lemmatizing the text further would have resulted in better text for training the models. Removing entries from the training set at the preprocessing stage might have also impacted the final result now that the model where trained on the original tweet text.

While the model is simple and fast, more advanced techniques might have further improved the classifiers. Experimenting further with ensemble methods besides Random Forest including GBoots might have proved a more fruitful pursuit but based on the initial behaviour I opted for the initially best-performing classifiers. These methods when properly optimized are proven to deliver F1 scores above 0.9 [1].

References

1. Jin, Z., Cao, J., Zhang, Y. and Zhang, Y., 2015, September. MCG-ICT at MediaEval 2015: Verifying Multimedia Use with a Two-Level Classification Model. In MediaEval.
2. Meyers, M., Weiss, G., Spanakis, G. (2020). Fake News Detection on Twitter Using Propagation Structures. In: van Duijn, M., Preuss, M., Spaiser, V., Takes, F., Verberne, S. (eds) Disinformation in Open Online Media. MISDOOM 2020. Lecture Notes in Computer Science(), vol 12259. Springer, Cham. https://doi.org/10.1007/978-3-030-61841-4_10
3. Taskin, S.G., Kucuksille, E.U. & Topal, K. Detection of Turkish Fake News in Twitter with Machine Learning Algorithms. Arab J Sci Eng **47**, 2359–2379 (2022). https://doi.org/10.1007/s13369-021-06223-0
4. Z Khanam, B N Alwasel, H Sirafi, and M Rashid 2021. Fake News Detection Using Machine Learning Approaches. IOP Conference Series: Materials Science and Engineering, 1099(1), p.012040.
5. Pedro Henrique Arruda Faustini, and Thiago Ferreira Covões 2020. Fake news detection in multiple platforms and languages. Expert Systems with Applications, 158, p.113503.
6. S. Krishnan and M. Chen, "Identifying Tweets with Fake News," 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 2018, pp. 460-464, doi: 10.1109/IRI.2018.00073.
7. Bharti, K.K., Pandey, S. Fake account detection in twitter using logistic regression with particle swarm optimization. Soft Comput **25**, 11333–11345 (2021). https://doi.org/10.1007/s00500-021-05930-y
8. Roshan, Rubab, Irfan Ali Bhacho, and Sammer Zai. 2023. "Comparative Analysis of TF–IDF and Hashing Vectorizer for Fake News Detection in Sindhi: A Machine Learning and Deep Learning Approach" Engineering Proceedings 46, no. 1: 5. https://doi.org/10.3390/engproc2023046005

Appendix 1.

Count of tweets for each confidence bin:

Confidence Bin	Count
(-0.001, 0.1]	998
(0.1, 0.2]	0
(0.2, 0.3]	4
(0.3, 0.4]	2
(0.4, 0.5]	98
(0.5, 0.6]	501
(0.6, 0.7]	30
(0.7, 0.8]	521
(0.8, 0.9]	901
(0.9, 1.0]	11222

Detailed breakdown with examples:

Confidence Bin: (-0.001, 0.1]
Language: err, Text:
Language: err, Text:
Language: err, Text:
Language: err, Text:
Language: err, Text:

Confidence Bin: (0.2, 0.3]
Language: et, Text: Ya se fue, ya paso Sandy,... koreko guaaaa .....jajajaja
Language: so, Text: Ok sandy..u win
Language: id, Text: Shhh... Oigan, ya se fue Sandy?
Language: de, Text: i am algerian BringBackOurGirls

Confidence Bin: (0.3, 0.4]
Language: af, Text: Idk man seems legit. XD
Language: no, Text: The Biggest Storm Ever!!!! LOL!!!!

Confidence Bin: (0.4, 0.5]
Language: pl, Text: nasagoddard's photo wow!
Language: pt, Text: Dear mainstream media. From Cuba.
Language: so, Text: Ok sandy..u win
Language: es, Text: La buena prensa inglesa (British tabloids) "at their finest..." All ya saben dnde est el desaparecido vuelo MH370.
Language: cy, Text: Dam you

Confidence Bin: (0.5, 0.6]
Language: es, Text: Superbe photo de la tempete pass a la sauce par original chez Pascal Roth
Language: cy, Text: HOLY SHIT nasagoddard's photo be careful
Language: id, Text: ne dah jmpa dah MH370 hahaha
Language: sk, Text: jodibiiitch24's photo
Language: af, Text: A birds eye view AC

Confidence Bin: (0.6, 0.7]
Language: it, Text: Impresionante foto del Huracn (va teflontara)

Appendix 1. Continued

Language: sv, Text: Hoboken NJ
Language: ca, Text: Un taur pels carrers de New Jersey
Language: pt, Text: asi se veia NY antes de que tocara tierra
Language: da, Text: Stormen Sandy skrmmer alla...

Confidence Bin: (0.7, 0.8]
Language: en, Text: Underwater bedroom at Poseidon Undersea Resort in Fiji. Who wouldn't like to live in a place like this! \n
Language: en, Text: I've never had more respect for someone.
Language: es, Text: Huracn Sandy arrastra 2 tiburones hasta el metro de New Jersey. Impactante.
Language: en, Text: 'Merica-1. Sandy-0. Fuck yeah.
Language: es, Text: El Empire State el nico iluminado.

Confidence Bin: (0.8, 0.9]
Language: af, Text: Sandy did not like trampolines ....
Language: en, Text: Soldiers:1 Sandy:0
Language: en, Text: Remarkable   cover (h/t via
Language: fr, Text: I'm out!
Language: en, Text: Lower Manhattan!!

Confidence Bin: (0.9, 1.0]
Language: en, Text: Praying for the city that I love and dream about going to.
Language: en, Text: Oh naw sharks in the streets
Language: en, Text: New species of fish found at Arkansas \/\/t.co\/E218nP6DZd
Language: en, Text: the Statue of Liberty is hiding from
Language: en, Text: WELCOME TO NYC SANDY.!!