SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

01.112 Machine Learning, Spring 2018

Design Project

Due 18 Apr 2018, 5pm

This project will be graded by Thilini Cooray
Please submit your work to eDimension.

Please form groups for this project early, and start this project early.

Instructions

Please read the following instructions carefully before you start this project:

- This is a group project. You are allowed to form groups in any way you like, but each group must consist of either 2 or 3 people. Please send your group information to eDimension latest by Friday 2 March 2018 5pm.
- You are strictly NOT allowed to use any external resources or machine learning packages. You will receive 0 for this project if you do so.
- Part 1 deadline is Monday 12 Mar 2018 5pm. **Please start working on part 1 as early as possible** as this part is done **individually**, and you do not need to form a team before you start. Annotated training and development set will be shared with you all on 19 Mar 2018.
- Each group should submit code together with a report summarizing your work, and give clear instructions on how to run your code. Please also submit your system's outputs. Your output should be in the same column format as that of the training set.

Project Summary

Many start-up companies are interested in developing automated systems for analyzing sentiment information associated with social media data. Such sentiment information can be used for making important decisions such as making product recommendations, predicting social stance and forecasting financial market trends.

The idea behind sentiment analysis is to analyze the natural language texts typed, shared and read by users through services such as Twitter and Weibo and analyze such texts to infer the users' sentiment information towards certain targets. Such social texts can be different from standard texts that appear, for example, on news articles. They are often very informal, and can be very noisy. It is very essential to build machine learning systems that can automatically analyze and comprehend the underlying sentiment information associated with such informal texts.

In this design project, we would like to design our sequence labelling model for informal texts using the hidden Markov model (HMM) that we have learned in class. We hope that your sequence labelling system for informal texts can serve as the very first step towards building a more complex, intelligent sentiment analysis system for social media text.

The files for this project are in the files `RU.zip`, `ES.zip`, as well as `SG.zip`, `CN.zip` (the latter two will be available on 19 Mar 2018, after we all have finished part 1). For each dataset, we provide a labelled training set `train`, an unlabelled development set `dev.in`, and a labelled development set `dev.out`. The labelled data has the format of one token per line with token and tag separated by tab and a single empty line that separates sentences.

The format (for the `SG` dataset, for example) can be something like the following:

```
Best O
Deal O
Chiang B-positive
mai I-positive
Tours I-positive
, O
The O
North O
of O
Thailand B-neutral
To O
Get O
special O
Promotion O
and O
free O
Transfer O
roundtrip O
. O
Contact O
: O
... O
http://t.co/sSn1OBTZ O

Independent B-neutral
Research I-neutral
Network I-neutral
News I-neutral
is O
out O
! O
http://t.co/KU5k7aHe O
! O
```

where labels such as `B-positive, I-positive` are used to indicate **B**eginning and the **I**nside of the entities which are associated with a positive sentiment. `O` is used to indicate the **O**utside of any entity. Similarly for `B-negative, I-negative` and `B-neutral, I-neutral`, which are used to indicate entities which are associated with negative and neutral sentiment, respectively.

Overall, our goal is to build a sequence labelling system from such training data and then use the system to predict tag sequences for new sentences. Specifically:

- We will be building two sentiment analysis systems for two different languages from scratch, using our own annotations.
- We will be building yet another two sentiment analysis systems for two different languages using annotations provided by others.

1 Part 1 (*15 points*, due 12 Mar 2018 at 5pm. Please budget your time well.)

The first and most important step towards building a supervised machine learning system is to get annotated data. This is also often one of the most difficult and most challenging steps in building a practical machine learning system, as we will see in this project. To allow each of us to have a full end-to-end experience on how challenging it is to build a practical supervised machine learning system, in the first part of this project, we will work together to get annotated data for performing sentiment analysis from social media data (some of them are collected from local social media users). You will receive 10 points if you complete the annotation. An additional 5 points will typically be awarded to you too unless we found the quality of your annotation is unacceptable. We will use an automatic approach to assess the quality of your annotations. Your annotations will be compiled and distributed to your fellow students in order to complete Part 2 and Part 3 of this project.

Please visit the following site for the annotation interface:

`http://ml-project.statnlp.org/annotation.php?id=/**YOUR_ID_HERE**/`

You then need to key in your student ID to proceed to the next step for annotation. For example, if your student ID is 1001949, please visit the following link for annotation:

`http://ml-project.statnlp.org/annotation.php?id=1001949`

Alternatively, visit the following site and type in your student ID:

`http://ml-project.statnlp.org/annotation.php`

You need to log in to start the annotation process. The user name and password are both your student ID. We also provide a sample collection of annotated data, which is available here:

`http://ml-project.statnlp.org/annotation.php?id=1000000`

Essentially, we are interested in annotating all major entities together with their sentiment information. Detailed instructions on the annotation can be found at *http://ml-project.statnlp.org/annotation.pdf*. The annotation interface is straightforward to use, but if you have questions there is a manual here:

`http://brat.nlplab.org/manual.html`

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

01.112 Machine Learning, Spring 2018 Design Project

Annotation Guidelines

27 Feb 2018

This is the annotation guideline for annotating named entity and sentiment present in a Twitter corpus and a Weibo corpus. If there is any question, please contact Thilini Cooray <thilini_cooray@mymail.sutd.edu.sg>.

According to the survey done previously, some of you are chosen to do annotations for Weibo, and some are chosen to do annotations for Twitter. Please refer to the web link described in project description to view your annotation contents.

The following pages contain two guidelines for Twitter and Weibo corpus respectively. Please read the guideline accordingly.

TWITTER CORPUS ANNOTATION GUIDELINE

1. INTRODUCTION

In this annotation task, we will annotate named entities and sentiment polarity towards each named entity in a Twitter Corpus. We will explain the definition of named entities and sentiment polarity in the following sections as well as the way to identify them.

2. NAMED ENTITY

2.1 Definition

According to definition in Wikipedia, in information extraction, a named entity is a real world object such as person, location, organization, product, etc., that can be denoted with a proper name. It can be abstract or have a physical existence. Examples of named entities include Barack Obama, New York, Volkswagen Golf, or anything that can be named. Named entities can simply be viewed as entity instances (e.g. New York is an instance of a city).

For the purpose of this annotation task, we need to annotate the outermost named entity phrases found in the Twitter corpus. Being the outermost noun phrase means it is not part of any larger noun phrases. For example, in the text "*I went to the Bank of China*". Note that we will tag *Bank of China* instead of *China*, since *China* is part of a larger named entity phrases.

2.2 Identification

A good method to determine if a target phrase is a named entity is to check if it is a name of **person, organization, location, product**. These four types will be entity types for selection during annotation. Note that, in one Twitter post, there may be several named entities to annotate.

2.3 Special cases

The following are named entities

- hashtags are names only if they are part of the sentence (e.g. not tacked on to the end).
- holidays ↙ How to annotate (254)(401)

The following are never named entities:

- dates and times
- seasons (e.g. "autumn")
- @-mentions
- addresses
- web trends like Throwback Thursday or Follow Friday

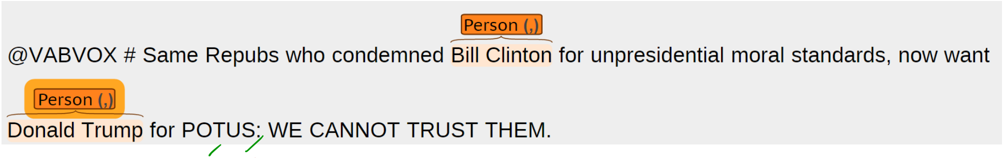
Don't annotate overlapping spans; if there are overlapping spans, choose the longer one.
Quotes surrounding an entity are not included in the span.

2.4 Example

Person

Latinos, Catholics (57) ✗
The Scottish lass (303) ✗
Clipper & Wef (310) ✗ → Org

A named person is tagged as Person type. Annotate "God" as a person. For example:

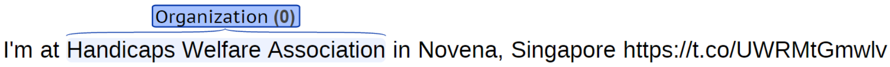

@VABVOX # Same Repubs who condemned Bill Clinton for unpresidential moral standards, now want Donald Trump for POTUS: WE CANNOT TRUST THEM.

Organization

✓ ✓
Band (228) (236)
afl (104) ✓ ✓ ✓
sports team? (35) (116)
1st People (38) ✗

Portillo's (97) ✗

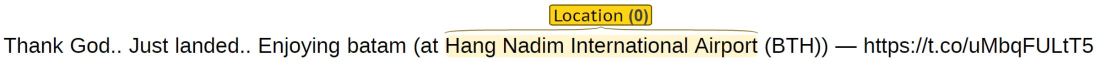
A named organization including company is tagged as Organization type. Websites are companies (Twitter, Netflix, etc…). For example:


I'm at Handicaps Welfare Association in Novena, Singapore https://t.co/UWRMtGmwlv

Location

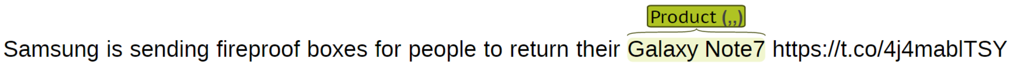
(441) ✗
Cervantas (89) ✗
Liverpool (92) ✗

A named location such as a named city, a named place is tagged as Location type. For example:


Thank God.. Just landed.. Enjoying batam (at Hang Nadim International Airport (BTH)) — https://t.co/uMbqFULtT5

Product (481) ✗

A named product such as software, hardware, movie, book, etc, is tagged as Product type. For example:


Samsung is sending fireproof boxes for people to return their Galaxy Note7 https://t.co/4j4mablTSY

3. SENTIMENT POLARITY

3.1 Definition

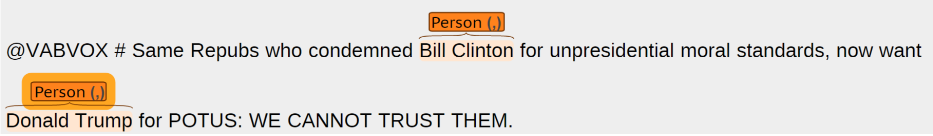
Sentiment Polarity is defined as sentiment expressed on the name entity target phrase. It has five types: **Very Positive, Positive, Neutral, Negative and Very Negative**, which are based on how annotator understands the sentiment information, mixed with subjective and objective information.

3.2 Identification

One way to determine the polarity depends how annotator feel about the target named entity based on understanding the sentiment information of sentence as well as personal subjective feeling. For example, in the sentence "*Stephen Curry is very popular in NBA now*", *Stephen Curry* and *NBA* are two named entities. Sentiment Polarity on *Stephen Curry* is positive because the sentence mentions he is very popular. Sentiment polarity on *NBA* is neutral, because it is an organization without much description in terms of sentiment.

3.3 Example

Sentiment Polarity is inferred from meaning of sentence. For example, *Bill Clinton* and *Donald Trump* are both tagged with *Negative* polarity.



@VABVOX # Same Repubs who condemned Bill Clinton for unpresidential moral standards, now want Donald Trump for POTUS: WE CANNOT TRUST THEM.

WEIBO CORPUS ANNOTATION GUIDELINE

1. INTRODUCTION

In this annotation task, we will annotate named entities and sentiment polarity towards each named entity in a Weibo Corpus. We will explain the definition of named entities and sentiment polarity in the following sections as well as the way to identify them.

2. NAMED ENTITY

2.1 Definition

According to definition in Wikipedia, in information extraction, a named entity is a real world object such as person, location, organization, product, etc., that can be denoted with a proper name. It can be abstract or have a physical existence. Examples of named entities include 爱因斯坦, 上海, 阿里巴巴, or anything that can be named. Named entities can simply be viewed as entity instances (e.g. 上海 is an instance of a city).

For the purpose of this annotation task, we need to annotate the outermost named entity phrases found in the Weibo corpus. Being the outermost noun phrase means it is not part of any larger noun phrases. For example, in the text "*我去中国银行*", note that we will tag *中国银行* instead of *中国*, since *中国* are part of a larger named entity phrases.

2.2 Identification

A good method to determine if a target phrase is a named entity is to check if it is a name of **person, organization, location, product**. These four types will be entity types for selection during annotation. Note that, in one Weibo post, there may be several named entities to annotate.

2.3 Special Cases

The following are named entities

- hashtags are names only if they are part of the sentence (e.g. not tacked on to the end).
- holidays

The following are never named entities:

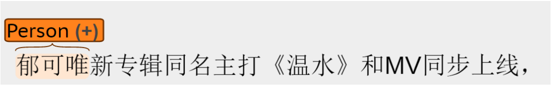
- dates and times
- seasons (e.g. "秋天")
- @-mentions
- addresses
- web trends like Weibo events

Don't annotate overlapping spans; if there are overlapping spans, choose the longer one.
Quotes surrounding an entity are not included in the span.

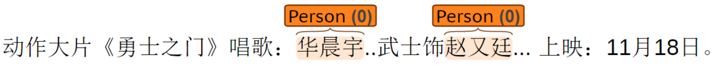
2.4 Example

Person

A named person is tagged as Person type. For example:



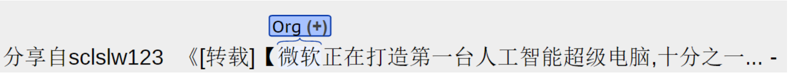
郁可唯新专辑同名主打《温水》和MV同步上线，



动作大片《勇士之门》唱歌：华晨宇..武士饰赵又廷... 上映：11月18日。

Organization

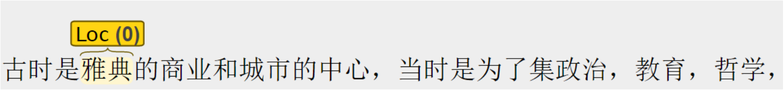
A named organization including company is tagged as Organization type. Websites are companies (新浪微博, 搜狐, etc…)For example:



分享自sclslw123 《[转载]【微软正在打造第一台人工智能超级电脑,十分之一... -

Location

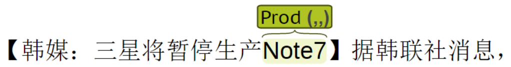
A named location such as a named city, a named place is tagged as Location type. For example:



古时是雅典的商业和城市的中心，当时是为了集政治，教育，哲学，

Product

A named product such as software, hardware, movie, book, etc, is tagged as Product type. For example:



【韩媒：三星将暂停生产Note7】据韩联社消息，

3. SENTIMENT POLARITY

3.1 Definition

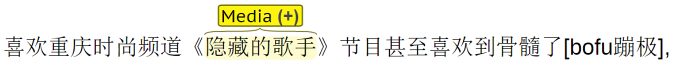
Sentiment Polarity is defined as sentiment expressed on the name entity target phrase. It has five types: **Very Positive, Positive, Neutral, Negative and Very Negative**, which are based on how annotator understands the sentiment information, mixed with subjective and objective information.

3.2 Identification

One way to determine the polarity depends how annotator feel about the target named entity based on understanding the sentiment information of sentence as well as personal subjective feeling. For example, in the sentence “*史蒂芬库里在 NBA 中非常受欢迎*”, *史蒂芬库里* and *NBA* are two named entities. Sentiment Polarity on *史蒂芬库里* is positive because the sentence mentions he is very popular. Sentiment polarity on *NBA* is neutral, because it is an organization without much description in terms of sentiment.

3.3 Example

Sentiment Polarity is inferred from meaning of sentence. For example, *隐藏的歌手* is tagged with *Positive* polarity.



喜欢重庆时尚频道《隐藏的歌手》节目甚至喜欢到骨髓了[bofu蹦极],

Disclaimer: to grant us the right to re-distribute your annotated data to your other fellow classmates and for potential future usage, by submitting your annotations online, you agree that your annotated data will be in public domain unless otherwise stated. Please contact Thilini Cooray if you have questions or doubts on this.

2 Part 2 *(25 points)*

Recall that the HMM discussed in class is defined as follows:

$$p(x_1, \dots, x_n, y_1, \dots, y_n) = \prod_{i=1}^{n+1} q(y_i|y_{i-1}) \cdot \prod_{i=1}^{n} e(x_i|y_i) \tag{1}$$

where $y_0 = \texttt{START}$ and $y_{n+1} = \texttt{STOP}$. Here q are transition probabilities, and e are emission parameters. In this project, x's are the natural language words, and y's are the tags (such as `O, B-positive`).

- Write a function that estimates the emission parameters from the training set using MLE (maximum likelihood estimation):

  $$e(x|y) = \frac{\texttt{Count}(y \to x)}{\texttt{Count}(y)}$$

  (5 points)

- One problem with estimating the emission parameters is that some words that appear in the test set do not appear in the training set. One simple idea to handle this issue is as follows. We introduce a special word token `#UNK#`, and make the following modifications to the computation of emission probabilities:

  $$e(x|y) = \begin{cases} \frac{\texttt{Count}(y \to x)}{\texttt{Count}(y)+k} & \text{If the word token } x \text{ appears in the training set} \\ \frac{k}{\texttt{Count}(y)+k} & \text{If word token } x \text{ is the special token } \texttt{\#UNK\#} \end{cases}$$

  (This basically says we assume from any label y there is a certain chance of generating `#UNK#` as a rare event, and empirically we assume we have observed that there are k occurrences of such an event.)

  During the testing phase, if the word does not appear in the training set, we replace that word with `#UNK#`.

  Set k to 1, implement this fix into your function for computing the emission parameters.

  (10 points)

- Implement a simple sentiment analysis system that produces the tag

  $$y^* = \arg\max_y e(x|y)$$

  for each word x in the sequence.

  For all the four datasets `RU`, `ES`, `CN`, and `SG`, learn these parameters with `train`, and evaluate your system on the development set `dev.in` for each of the dataset. Write your output to `dev.p2.out` for the four datasets respectively. Compare your outputs and the gold-standard outputs in `dev.out` and report the precision, recall and F scores of such a baseline system for each dataset.

The precision score is defined as follows:

$$\texttt{Precision} = \frac{\text{Total number of correctly predicted entities}}{\text{Total number of predicted entities}}$$

The recall score is defined as follows:

$$\texttt{Recall} = \frac{\text{Total number of correctly predicted entities}}{\text{Total number of gold entities}}$$

where a gold entity is a true entity that is annotated in the reference output file, and a predicted entity is regarded as correct if and only if it matches exactly the gold entity (*i.e.*, both their *boundaries* and *sentiment* are exactly the same).

Finally the F score is defined as follows:

$$\texttt{F} = \frac{2}{1/\texttt{Precision} + 1/\texttt{Recall}}$$

Note: in some cases, you might have an output sequence that consists of a transition from `O` *to* `I-negative` *(rather than* `B-negative`*). For example, "*`O I-negative I-negative O`*". In this case, the second and third words should be regarded as one entity with negative sentiment.*

You can use the evaluation script shared with you to calculate such scores. However it is strongly encouraged that you understand how the scores are calculated.

(10 points)

3 Part 3 *(20 points)*

- Write a function that estimates the transition parameters from the training set using MLE (maximum likelihood estimation):

  $$q(y_i|y_{i-1}) = \frac{\texttt{Count}(y_{i-1}, y_i)}{\texttt{Count}(y_{i-1})}$$

  Please make sure the following special cases are also considered: $q(\texttt{STOP}|y_n)$ and $q(y_1|\texttt{START})$.

  (5 points)

- Use the estimated transition and emission parameters, implement the Viterbi algorithm to compute the following (for a sentence with n words):

  $$y_1^*, \ldots, y_n^* = \arg\max_{y_1,\ldots,y_n} p(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

  For all datasets, learn the model parameters with `train`. Run the Viterbi algorithm on the development set `dev.in` using the learned models, write your output to `dev.p3.out` for the four datasets respectively. Report the precision, recall and F scores of all systems.

  Note: in case you encounter potential numerical underflow issue, think of a way to address such an issue in your implementation.

  (15 points)

4 Part 4 *(20 points)*

- Use the estimated transition and emission parameters, implement an algorithm to find the 5-th best output sequences. Clearly describe the steps of your algorithm in your report.

  Run the algorithm on the development sets `RU/dev.in` and `ES/dev.in` only. Write the outputs to `RU/dev.p4.out` and `ES/dev.p4.out`. Report the precision, recall and F scores for the outputs for both languages.

  Hint: find the top-5 best sequences using dynamic programming by modifying the original Viterbi algorithm.

  (20 points)

5 Part 5 – Design Challenge *(20 points)*

- Now, based on the training and development set, think of a better design for developing an improved sentiment analysis system for tweets using any model you like. Please explain clearly the model/method that you used for designing the new system. We will check your code and may call you for an interview if we have questions about your code. Please run your system on the development set `RU/dev.in` and `ES/dev.in`. Write your outputs to `RU/dev.p5.out` and `ES/dev.p5.out`. Report the precision, recall and F scores of your new systems for these two languages.

  (10 points)

- We will evaluate your system's performance on two held out test sets `RU/test.in` and `ES/test.in`. The test sets will only be released on 16 Apr 2018 at 5pm (48 hours before the deadline). Use your new system to generate the outputs. Write your outputs to `RU/test.p5.out` and `ES/test.p5.out`.

  The system that achieves the overall highest F score on the test sets will be announced as the winner.

  (10 points)

Hints: Can we handle the new words in a better way? Are there better ways to model the transition and emission probabilities? Or can we use a discriminative approach instead of the generative approach? Perhaps using Perceptron?[1]. *Any other creative ideas? Note that you are allowed to look into the scientific literature for ideas.*

Items To Be Submitted

Upload to eDimension a single ZIP file containing the following: (Please make sure you have only one submission from each team only.)

- A report detailing the approaches and results
- Source code (.py files) with README (instructions on how to run the code)
- Output files

[1] http://www.aclweb.org/anthology/W02-1001

- `RU/`
  1. `dev.p2.out`
  2. `dev.p3.out`
  3. `dev.p4.out`
  4. `dev.p5.out`
  5. `test.p5.out`
- `ES/`
  1. `dev.p2.out`
  2. `dev.p3.out`
  3. `dev.p4.out`
  4. `dev.p5.out`
  5. `test.p5.out`
- `CN/`
  1. `dev.p2.out`
  2. `dev.p3.out`
- `SG/`
  1. `dev.p2.out`
  2. `dev.p3.out`