PGM - Assignment 3 Part B
Akshay K Gupta - 2013CS50275

Q1. I modified a little bit of Mallet code to make it converge quicker.
In addition to the basic POS and NER models I trained the models with the orders option set to 2. Order more than 2 was taking too much time and running out of heap space. The accuracies are as follows:

Model	Macro F-score	Macro Precision	Macro Recall
Basic POS Model (Order = 1)	0.606897	0.669608	0.554926
Basic POS Model (Order = 2)	0.513112	0.585231	0.456817

Model	Macro F-score	Macro Precision	Macro Recall
Basic NER Model (Order = 1)	0.173822	0.556373	0.103001
Basic NER Model (Order = 2)	0.125920	0.425833	0.073884

Observations :

1. Increasing the order leads to a sharp decrease in performance. This is unexpected because a second Order Markov assumption gives a more expressive model. The reason could be overfitting or the highly irregular structure of tweets.
2. The F-score of NER is very low because the precision and recall of the labels 'Facility' and 'Company' are particularly low.
3. The basic POS model does not recognise URLs and AT-mentions effectively.

Q2. I modified the code of SimpleTagger.java to create an HMM model instead of a CRF one. It trains successfully on the training data. However, when exposed to a new test set, it cannot label it because it sees new words which did not occur in the training data, and as HMM models joint over words and tags, it cannot handle these new words.

Q3. For the POS models the features I used (incrementally) were:

1. Capital (CAP): This feature is on if the first letter of the word is capital. The intuition is to help the model tag proper nouns (^) more effectively.
2. Ending with -ing (ING): This feature is on if the word ends with -ing. The intuition is to help the model tag verbs (V) more effectively.
3. @-mention (AT): This feature is on if the word is an @-mention. The intuition is to help the model tag @-mentions (@) more effectively.
4. Hashtag (HASH): This feature is on if the word begins with a hashtag. The intuition is to help the model tag hashtags (#) more effectively.
5. Ending with -ly (LY): This feature is on if the word ends with -ly. The intuition is to help the model tag adverbs (R) more effectively.
6. Ending with -'re or -'m (NV): This feature is on if the word ends with 'm or 're. The intuition is to help the model tag Nominal+verbal words (L) more effectively.
7. URLs (URL): This feature is on if the word begins with 'http'. The intuition is to help the model tag URLs (U) more effectively.

8. Ending with ’s (POSS): This feature is on if the word ends with -’s. The intuition is to help the model tag possessive words (S, Z) more effectively.
9. Similar phonetic words: The Double Metaphone algorithm is used which labels similar sounding words as a compressed word. This helps group words like (thanks, thanx, thankss - TNKS) together.

The accuracies obtained are as follows:

Feature	Macro F-score	Macro Precision	Macro Recall
None	0.606897	0.669608	0.554926
CAP	0.624202	0.681537	0.575765
ING	0.612802	0.654898	0.575791
AT	0.643332	0.691308	0.601582
HASH	0.668716	0.718968	0.625031
LY	0.670084	0.720019	0.626626
NV	0.669728	0.714991	0.629855
URL	0.690780	0.739558	0.648037
POSS	0.703977	0.737016	0.673773
Phonetic Grouping	0.715209	0.763036	0.673025

Observations:
1. The accuracies mostly increase except for when ING and LY features are added.
2. The best model is a combination of all of the above features, which matches expectations.
3. The F-score is hit because for some of the low occurring labels (X, S, Z), the precision and recall values are very low, which means the models is not able to identify these labels.

For the NER model the features I used were:
1. Capital (CAP): This feature is on if the first letter of the word is capital. This feature helps separate the O label from every other label.
2. POS tag: POS tag for each word (as labelled by the best POS tagger above) is added as a feature.
3. POS tag for only proper nouns: The POS tag is added only if the tag is proper noun (^) because the main goal of POS tagging is to try and separate proper nouns from the rest. This is tested separately from the previous feature, not incrementally.
4. Place (LOC): A lookup table of places in the world is used and if the word or (word + next word) matches any of these places then this feature is on. The intuition is to try and explicitly tell the model what the locations (geo-loc) are. The list of places is taken from a UN site.

The accuracies obtained are as follows:

Feature	Macro F-score	Macro Precision	Macro Recall
None	0.173822	0.556373	0.103001
CAP	0.241976	0.510105	0.158606
POS Tag	0.178012	0.332903	0.121488
Proper Noun tag	0.232137	0.346890	0.174433
LOC	0.215666	0.303393	0.167293

Observations:

1. Adding the CAP feature causes a good performance increase but all the other features cause a decrease in performance.
2. Decrease in case of proper noun may be because the proper noun tagging is not being very well, which is because the precision and recall of the POS tagger for proper nouns is around 0.5 each.
3. Decrease in case of the LOC feature may be because too many things are being labelled as LOC including persons, companies and facilities if their name matches a place name. The probability of a person, company or facility matching a place name is not low because the list of places used is extensive.