First, the author preprocesses the candidates.txt by adding two criteria. One is a word should not have more than 3 consecutive repetitive letters. Another is a word length should not be longer than 30. Second, the author uses Local Edit Distance (distance>3) to identify blends, and the result is 1.23% precision with 80.39% recall. Third, the author uses Jaro-Winkler Similarity (similarity >= 0.91) to identify blends, and the result is 1.25% precision with 71.89% recall. And, the author concluded that Local Edit Distance is slightly better than Jaro-Winkler Similarity in this project because of higher recall. Finally, the author made conclusions and had some suggestions.

First, the author did some preprocessing of candidate words and I think the assumption is reasonable. Words either too long or have many consecutive repetitive letters are excluded. Second, two methods (Local Edit Distance and Jaro-Winkler Similarity) were used to identify blend words and the results are convincing. The author compared the results and made some suggestions. The examples of identified right or wrong blends are given. Third, the report has good introduction, which helps the target audience to understand the project. Finally, the author found the limitation of the approach and had some reflections.

First, I think the author could do more in the preprocessing. For example, a word has non-consecutive repetitive letters (e.g. ‘hahaha’) cannot be a blend as well. Second, I think the author could explain more about why (s)he chose distance>3 in Local Edit Distance and similarity >= 0.91 in Jaro-Winkler Similarity (Are they based on the project results or any studies?). Third, I think using one string metrics only is hard to achieve good results, so the author may consider using other methods with string metrics to improve the results. Finally, in the conclusion part, the author mentioned a lot about the Twitter dataset. I think maybe they should be moved to the introduction part because I think the conclusion is meant for results, reflections, and suggestions.

============================================================

First, the author preprocesses the candidates.txt by adding four restrictions. (1) Length of blends >= 4. (2) Length of candidates > 2. (3) Prefix or suffix letters >=2. (4) Prefix is not suffix. After this step, precision is 1.41% and recall is 83.44%. Second, the author uses Neighborhood Search to get rid of the misspelled word. Control Experiment is used to compare results on “no restrictions” / “restrictions” dataset. Precision is 1.73% / 2.219% and recall is 58.94% / 39.10% with 1 edit, and precision is 2.35% / 3.49% and recall is 50.33% / 30.46% with 2 edits. Third, based on “no restrictions with 1 edit”, Global Edit Distance is used to identify blends, and the result is not ideal. Finally, N-gram Distance is used and improved the result (1.57% precision and 66.23% recall). Combined all, the final result is 2.62% precision with 47.68% recall.

First, the logic and design of this project are clear. (1) Use “restrictions” to exclude non-blends. (2) Use Neighborhood Search to exclude misspelled words. (3) Use Global Edit Distance or N-gram Distance to identify blends. Second, controlled experiments are well designed to show the different results in the project. We can see the comparison between “restrictions” and “no restrictions” and the comparison of different thresholds in the Edit Distance and N-gram. Third, analyses are convincing. When recall decreases after using Neighborhood Search, reasons are explained the and examples are given about what true blends got excluded. Finally, the overall report quality is good. Graphs are intuitive and analyses are reasonable.

First, in section 4.3, there are four tables. I think they can be put into one, which would be more intuitive. Second, the author mentioned Global Edit distance is not as good as N-gram Distance because of the results got show little benefits, but I think the author could explain a bit more. From my understanding, the GED is not suitable because, unlike Jaro-Winkler or N-gram, it only cares about how many edits needed, and does not put weight on prefix. Finally, in the conclusion part, I wish the author talks more about possible improvements.

First, the author analyzed real blend words and candidate words and got some good insights about the attributes and patterns of them. Second, the author designed 7 heuristics functions to exclude non-blend words, which includes (1) 4 <= Word length <= 13 (2) No consecutive or non-consecutive repetitive characters. (3) No exact match of Prefix and Suffix. (4) Typo detection with Jaccard Similarity and Jaro-Winkler Distance (5) Prefix or Suffix contribution >=2 letters (6) No same word stem (7) No rare Suffix. Third, the author used regular expressions (^prefix\w+, \w+suffix$) to find all the possible blends. The Result is that heuristic functions improved precision but reduced recall.

First, the strategy used in project is clear and reasonable, which is finding blends using regular expression and excluding blends with heuristic functions. Second, there are some good designs of heuristic functions in the project, which are useful in excluding non-blends. Some heuristic functions excluded many non-blends with 92% recall, while others improved precisions. Third, in the Typo Detection part, I think it is really effective to use Jaccard Similarity with Jaro-Winkler Distance to excluded misspelled words, and word stem may be also useful to decide if two words are the same. Finally, the results are convincing and there are some good reflections and suggestions about the improvements of this project.

First, the project used 7 different heuristic functions, but whether one would affect another was not determined. For example, heuristic 1 (4 <= Word length <= 13) might affect the results of Jaro-Winkler Distance. So control experiments could be designed to see their relationships. Second, using regular expression only is difficult to achieve high precision. So combine regular expression with other method, such as N-gram or statistical model, may improve the results. Finally, the author could explain the results a little bit more in the result section.

============================================================

First, the author analyzed real blend words and candidate words and got some good insights about the attributes and patterns of them. Second, the author designed 7 heuristics functions to exclude non-blend words, which includes (1) 4 <= Word length <= 13 (2) No consecutive or non-consecutive repetitive characters. (3) No exact match of Prefix and Suffix. (4) Typo detection with Jaccard Similarity and Jaro-Winkler Distance (5) Prefix or Suffix contribution >=2 letters (6) No same word stem (7) No rare Suffix. Third, the author used regular expressions (^prefix\w+, \w+suffix$) to find all the possible blends. The Result is that heuristic functions improved precision but reduced recall.

First, the strategy used in the project is clear and reasonable, which is finding blends using regular expression and excluding blends with heuristic functions. Second, there are some good designs of heuristic functions in the project, which are useful in excluding non-blends. Some heuristic functions excluded many non-blends with 92% recall, while others improved precisions. Third, in the Typo Detection part, I think it is really effective to use Jaccard Similarity with Jaro-Winkler Distance to excluded misspelled words, and word stem may be also useful to decide if two words are the same. Finally, the results are convincing and there are some good reflections and possible future improvements in this project.

First, the project used 7 different heuristic functions, but whether one would affect another was not determined. For example, heuristic 1 (4 <= Word length <= 13) might affect the results of Jaro-Winkler Distance. So control experiments could be designed to see their relationships. Second, using regular expression only is difficult to achieve high precision. So combine regular expression with other methods, such as N-gram or statistical model, may improve the results. Finally, the author could explain the results a little bit more in the result section.

============================================================

1

First, the author explored the dataset and chose to use BestXX set for the project since the features in MostXX do not indicate a class. Second, the author chose to use Accuracy, Precision and Recall to evaluate the effectiveness of the models. Third, the author chose Zero-R (baseline), Naive Bayes, J48 Decision Tree and Support Vector Machines (SVM) classifier to build the models. Fourth, the author got the results for each model and conclude that “their performance is not good enough” since the precision is just slightly higher than the baseline. Finally, the author proposed some ideas for future study. One is to remove words manually form the BestXX dataset to improve the result. Another is to use K-NN for this project.

2

First, the author explained the dataset used in this project in detail and gave some good examples in the report, which are very intuitive. Second, the author explained each machine learning algorithm used in this project and had some ideas about the strength and weaknesses of each model beforehand. I think this is very important before doing the experiment. If we have some ideas about the model, we are more likely to choose the models that are suitable for our projects. Third, the results are clearly shown in the table and the data are convincing. Fourth, the author analyzed the results and had some good reflections and ideas for future improvements.

3

First, table 2 (6.1 Zero-R, baseline) is a little bit redundant. Since Zero-R always predicts the majority class, which does not use any features in the dataset, best10 to best200 are not necessary. Second, the author mentioned building models for SVM and J48 are very slow if we have many features, which is a disadvantage. I think it is not technically wrong, but in real life, it is quite common to model a model using several days or even weeks. Third, I think the author could talk more about over-fitting problems in the analysis, especially for the J48 Decision Tree model. Fourth, I think the author could do more feature engineering in this project. Since there are many ‘0’s in our BestXX dataset, which is not ideal for building models. If the author can do something in this part, I think the results would be much better.

========================================================

First, the author did feature selections of the training set by using the Vader package to check the sentiment of a given tweet. The author also removed some common words in the dataset and recorded some attributes of a tweet, such as word counts. Second, the author decided to use a hold-out strategy (70:30) on the training set (96585 instances). Third, the author used Decision Tree, Naive Bayes, and K-NN to build the model. Fourth, the author used the YAKE package to find the best terms in tweets and run the model again with Voting Classifier. Fifth, the author compared the results of two experiments and concluded that the updated system is better due to the proper feature selection. Finally, the author had some ideas about future improvements.

First, the idea of using sentiment (emotion) as a feature of a given tweet is interesting. I think the assumption is that people in one city are happier than people in another city so they would post more positive comments. Second, the method (DT, NB, KN, and VC) and strategy (Hold-out) used in the project are well explained and the structure of the experiments is clear (One has a further selection while another has not). Third, the results are convincing and the author had some good reflections and ideas for future improvements. Fourth, the author tried applying the model to another dataset (IRIS by Weka). I think it is a good idea to see if our model works on some previously tested datasets.

First, although the author used the first experiment result as a baseline, I think the author should use some baseline classifiers such as Zero-R or One-R, otherwise, we would never know how much improvements our model has. Second, the author used a Hold-out strategy (70:30) on the training set. I think the author should use the provided development set for testing so we would have more training instances. Or the author can combine the training set and development set and use K-Fold Cross-Validation. Third, the author mentioned about python packages and how to import them into the project. I think it is too technical for the report so I recommend removing them. Fourth, I think the author could organize the report a little bit more. The number of decimals should be the same in the table and the hyperlink in the reference part should be formatted as well.