Contents

WS 2016 Project 1

Martin Simon Haugaard
cdl966@alumni.ku.dk

1. INTRODUCTION

The scope of this project it to try and create a prediction model which, based on Google term history, aims to predict the real-life sales of vaccines. The terms are extracted by short layman descriptions from two danish websites sundhed.dk and ssi.dk, which contain various information regarding public health. The terms extracted from these two websites were then parsed through Google Trends[1] resulting in relative search history regarding each term, this allowed for constructing a model in order to predict search trends for words regarding each vaccine, which may prove useful in real world sales predictions.

The following project was all done in python, using premade modules to assist in the mining and processing of trends.

In this project, the vaccines **PVC** & **HPV** were examined. Any two vaccines should be fine, but I'm happy with these two, as their data differs in size, with PVC only creating 16 query items, while HPV has 65. Query items are also called Trends, and are described in the following section.

2. METHODOLOGY

This section contains brief descriptions of the key areas of the project.

2.1 Trends

In this project, trends are words which are commonly used when describing vaccines. Stopwords, which are words that only appear in text for grammatically reasons, do not count as trends. The trends are determined by reading multiple (in this case two) descriptions of a given vaccine, and then stripping each description of symbols and stopwords. Both descriptions are then compared to each other, and only words which occur in both descriptions qualify to be trends. This is a rather simple way of creating the trends, and some words, which only differ grammatically, such as *alvorlig* and *alvorlige* are considered different, even through they are functionally the same. However it proves sufficient for the scope of this project.

[1] https://www.google.dk/trends/

Web Science 2016 DIKU, Denmark

2.2 Mining of Trends

In order to mine the trends from Google, I resorted to using a python package called pytrends[2], which provides for a simple framework for downloading .csv files from Google Trends. It works by signing into Google, and then, with different intervals, making a request after which it allows for download of the corresponding .csv file.

When downloading the .csv files, they can come with data in either a weekly or a monthly interval, most likely due to the popularity of certain terms, which prompts for more indept information. In order to accommodate for the difference of format, a simple python script was written, which would convert weekly data into monthly data, by making a crude estimation of a monthly value, based on average values and full week durations, which meant some weeks overlapped into the following month. A more advanced script should probably be written for a more precise estimation of monthly values, but for this project, I found the crude script to work well.

Certain trends may result in empty .csv files. For example, for PVC, *vaccinen*, *pneumokoksygdom* & *pneumokokvaccinen* came up empty, thus reducing the usable number of trends from 16 to 13. In these cases I simply skip the trend when doing further analysis.

The data values given in each .csv file will be a value in the interval [0, 100], being relative measurements of the popularity of the trend for a given period of time, 100 being the most popular interval during the measured time scope. This means that a trend scoring highly doesn't necessarily mean it was a highly searched term on Google overall, but rather, it's relatively highly searched compared for other time intervals regarding the same search term.

Lastly, it's worth noting that the trends mined were only for the Danish region of Google Trends, as the main scope for this project is the danish market.

2.3 Prediction

The choice of prediction method was randomly chosen to be that of Lasso. Lasso requires a Matrix, X, of dimensions NxM where N is the time frame we examine, here 57 months, and M is the number of trends (13 and 61 after empty results have been removed). The provided data served as a ground truth when preforming fitting of the model, and even

[2] https://github.com/GeneralMills/pytrends

through it holds data for 60 months, I only use the first 57, as to match the X Matrix.

Fitting of the model was done as part of a 5-fold cross-validation. The python package *sklearn*[3] provided all the necessary tools for doing the splitting needed in the 5-fold cross-validation, as well as the model fitting and prediction.

```
k_fold = cross_validation.KFold(len(X), 5)
```

would provide a *k_fold* consisting on five iterations on form $k, (train, test)$, where train is list of index values for training, and test is a smaller list for testing.

These values would allow me to gradually train the module, results of which are in the next section.

3. FINDINGS

First I examined PVC. Figure 1 shows how my predictions for each of the five folds went. Looking at these 5 folds, the

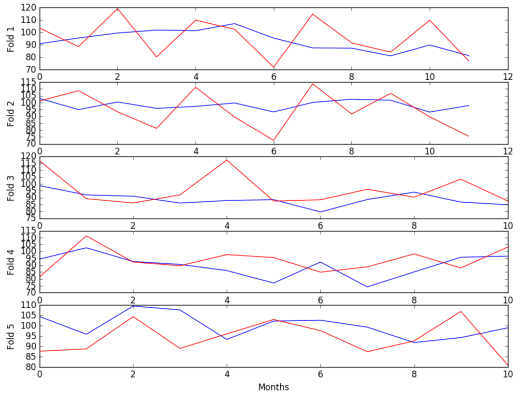

Figure 1: PVC predictions done in 5-fold cross-validation, red line is ground truth, blue is prediction.

predictions being made, in blue, does generally follow the provided data, in red, however not every fold is as spot on as fold 5. In figure 2 the prediction are is spot on Fold 2, as it predicts the sudden drop at 6 months, and then it practically follows the curve from there.

Other than simply looking at graphs, the way to determine how well a prediction model is doing, is by calculating the *Root-mean-square error*, or RMSE $= \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$, with n being the number of folds and e_t being the difference, or error, between the prediction and the real-life observed data for the corresponding fold. RMSE is useful for evaluating the accuracy of a model, as a low value indicates a very too correlation, as the error is generally low, and a higher value means a bigger average error.

In table 1 I've provided the average RMSE over all five loops, for each vaccine, when looking at each of the provided clinical data files. Next to my results, I've entered the results which the original report[1], which this project was based on, was able to achieve.

[3] http://scikit-learn.org/

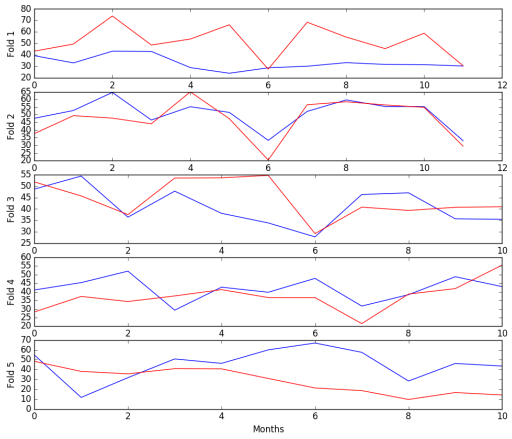

Figure 2: HPV predictions done in 5-fold cross-validation, red line is ground truth, blue is prediction.

For RMSE, the bigger the value you get, the worse your prediction probably is, and it's clear from the table, that my predictions were strictly worse than those provided in the original report, leading to the conclusion that my model is not quite as good.

Table 1: Lasso RMSE comparison, with Prediction Vaccination Uptake using Web Search Queries report.

	Original Report	My Findings
HPV-1	12.701	17.07
HPV-2	18.423	23.202
HPV-3	23.074	24.436
PCV-1	7.845	12.685
PCV-2	9.770	14.448
PCV-3	10.368	17.367

4. CONCLUSIONS

Examining table 1 it's clear that my predictions did not preform as well as those in the original report. Especially not for HPV. I find the HPV to be interesting through, as the clinical data for the last 11 months are drastically lower than the previous 49 months, as plotted in figure 3. The model tries to predict a curve, as close to the previous 49 months, but since there's a sudden change, it fails to predict the vaccine frequency.

As for PVC, figure 4, the prediction is hard to compare to the clinical data. For each spike it predicts, it fair to say it misses one as well. This may be due to the, compared to the HPV vaccine, there's a limited amount of trends, 13 versus 61, and while the RMSE is not too far off, compared to the original paper, the actual usefulness of the current state of the predictions are very risky, should one try to invest based on these data point.

Reason may be that the terms which were mined using

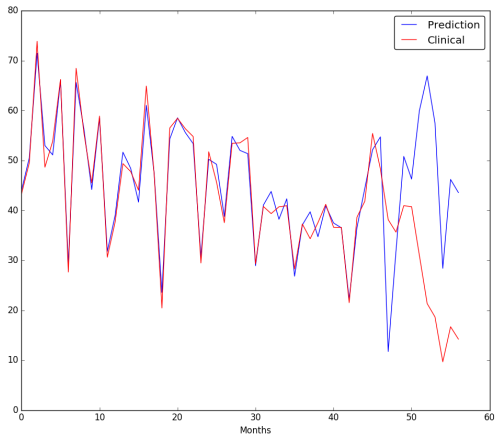

Figure 3: HPV Model predicted over full duration of data sample. red line is ground truth, blue is prediction.

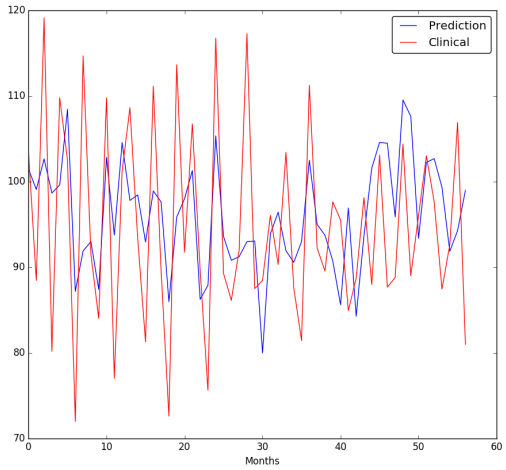

Figure 4: PVC Model predicted over full duration of data sample. red line is ground truth, blue is prediction.

Google all have equal influence on the prediction model, as they are all relative measurements, which gives no indication on the most important terms. Some terms may be irrelevant or misleading, as they can be too general for this specific area of interest.

Overall, the exercise was engaging and a learning experience. But it's hard to be sure of the exact usefulness of the models as is.

5. REFERENCES

[1] N. D. Hansen, C. Lioma, and K. Molbak. Predicting vaccination uptake using web search queries. *in press*, 2016.

Web Science
Peer Assessment

Peer Assessment

In assessment of the provided project, each section will be assessed individually with individual criticisms, and then the project as a whole will be assessed.

1. Introduction:

The introduction is strong, but succinct, almost too much so. I would argue that a bit more might strengthen the reader's understanding of what the project was about and some more background information about data itself and the vaccinations. It does not say that the data comes from the Google Trend data for Denmark which I think is an important distinction (as opposed to worldwide or another country). The use of personal pronoun I is another mechanical criticism albeit less important. Otherwise, a good, concise introduction where the vaccines are clearly identified and the project is successfully overviewed to introduce the reader to the subject.

2. Methodology:

a. Trends:
This section is well written with a clear description for "stopwords" as well as some valid self critique. Methods for stripping punctuation are left out (Not positive or negative just something I noticed). The writer clearly understands the academic level of the project and does not waste words or the reader's time rambling unnecessarily.

B. Mining of Trends
This section is again, very compact, which is good but misses a couple key points about the nature of the data. There is no mention of the fact that the data gathered is not absolute data but relative data, Google Trends compares the queries against each other not against a global measurement index. And again the personal I usage here is probably not applicable to an academic paper. Otherwise, the section is neatly organized, with consecutive tasks described quickly and effectively and saves space for possible room for error (a more advanced script, and poor trend data criticisms are covered).

C. Prediction:
Clearly describes and annotates the libraries used for prediction. Perhaps the summary of lasso and k fold cross validation were a bit excessive, but nonetheless relevant to the project (I would assume the reader would understand these concepts beforehand).

3. Findings:

The graphs provided were applicable and relevant to the project. The writing is very compact, again, I would say more explanation might be helpful to the reader. For example the actual frequency data was left out of the findings, which might help the reader understand the project more clearly. I found it easy to follow, but those that have not taken the subject in a long time might find it more difficult, especially with regards to the root mean square error. Overall, the section clearly and effectively presents the data in an understandable way.

4. Conclusions.

Here the writer puts all the conclusive evidence towards an assessment of the predictive model, yet there is little expansion upon the connection between the predictive dataset and the observed dataset. It simply says “But it's hard to be sure of the exact usefulness of the models as is.” Which is true but does not explain why this is the case other than stating that the error is larger in the predictive model. I would say that more here would strengthen the project, especially when elaborating on the connection between the frequency output from google and the relevancy towards a predictive model. The writer shows a clear understanding of the methodology and concepts around the project, but I believe a critical thinking (i.e. what do the predictions mean? How do outliers affect the predictive model? What other factors might be involved?) section might be helpful in expanding upon the findings of the data.

Overall I thought it was a very good project, the writer clearly understands what he/she is doing and how the data is taken and analyzed. However, I think more expansion upon the findings and perhaps connections between data could strengthen the project enormously. Again, elaborating upon the connection between the mined and observed data is one area for improvement. Otherwise, it is cleanly and effectively presented, with well labeled graphs and tables. The english is very easily understood and the ideas are relevant and helpful. I thought it was a very well done project.

Amendment List

Extend introduction to strengthen reader's understanding

The Introduction was extended to mention the websites used, and is not a short recap of the entire project in a single section.

Explain Google Trends data for Denmark

While intuative to know, I've added a brief sentence covering this, in section **2.2 Mining of Trends**.

Personal pronoun *'I'* not applicable academic paper

I can see the logic behind this claim, but I've chosen to keep using 'I', as it's a one person project, and using other pronouns such as "we" or "our" may confuse the reader into thinking it was a group effort, and avoiding any such pronoun altogether would result in hard to read language. Also, I've stayed true to using 'I' throughout the project, which is my main concern, keeping it continuous.

Trends - Missing methods for stripping punctuation

I find this trivial, and hence not included in a paper. I've made clear that I'm only interested in words as trends, hence it should be obvious that special characters are stripped.

Mining of Trends - Nature of Data Explanation, Not absolute but relative

Also worth noting, and as such a paragraph has been introduced in the section in question.

Summary of lasso and k-fold cross validation bit excessive.

I don't think it's too excessive to explain these points, as it's the main feature of the project. I would argue that the reader has an academic background, but k-fold and lasso (which I've only mentioned by name) was new to me at the beginning of this course, thus worth including.

Findings - Writing too compact, more explanation might be helpful

In order to accommodate this, I've expanded slightly on RMSE, as I find it to be the least intuitive part of what I've written.

Little expansion upon the connection between the predictive dataset and observed dataset

I find the connection given in Section **4. Conclusions** to be sufficient.

Elaborate on the connection between the frequency output from Google and the predictive

I've expanded slightly in the conclusions to explain that not every term may weigh equally in the real world, but they all have the same weight in the model.

Critical thinking section might be helpful

Changes made in conclusion section now serves as critical thinking as well.

More expansions upon the findings could strengthen the project enormously

As before, the findings / conclusions section has been expanded as pr request from other amendments, so I find this fulfilled.

Peer Review - Project One

Anonymous

March 16, 2016

Peer-assessment

First I'd give some general feedback, and then I will quickly give feedback on each of your sections.

General Remarks

First of all, use the latex template. It will help structure the report, which is badly needed.

The language used in the report should also be revised, for example there's a constant change between *we* and *I*, pick one and stick to it. Remember, you're writing an article. Think of it not as a first draft, neither as a short recab for a fellow student, but rather as a piece which you aim to have published, and have read by people outside this couse. This is why proper grammar, precise language and a strong structure is key, it will all help the reader to better understand and follow what you're trying to say.

When including data from your code, I strongly suggest you put these into either a table, a list or a figure, table and figure being described in the template. It allows for easy reference, while also making it clear for the reader that the data is special. Also, use references when mentioning foreign tools or reports.

Correctness of report

Other than having only computed values for one vaccine, the method explained in the report seems sound, although a bit lacking, as RMSE alone is a weak result.

Other general remarks

It does appear that the report covers all the essential parts of the requirements.

Certain parts of the report seemed to be more like filler comments than actual useful observations, keep descriptions short and to the point.

I'm missing examples and conclusions upon the examples. I don't see many choices being made, and it seems like you're just working towards a goal, without stopping and thinking of why and how to best get there.

As a fellow student, it's easy to understand what you are doing, but if the report were to be presented to a random computer science student, I am not sure they will get what it going on, there's too much you assume the reader already knows about.

Report feedback

Intro

Remember, it's no guarantee that the reader is a fellow student for this course. Try and quickly cover the motivation for why we are doing this.

Methodology

Finding queries

The section on how you found your list of queries is too long, and feels like an unnecessary summary. It can easily be summarized as taking descriptions of vaccines, and stripping them of unnecessary stop words and punctuations, after which descriptions for a certain vaccine was compared, and words extracted. It's far more important that you cover the general method of processing the data, rather than going into detail of everything you did, for example creating a file from another file, creating exactly four lists, appending elements to lists and using certain functions are hardly relevant for the general reader.

I don't see the relevance of ordering alphabetically.

Why are numbers excluded from the queries? Is it because they are very general? If so, can't certain other words may also be removed. I would like a quick reasoning as to why.

Getting frequencies

When you mention something new, like PyTrends, it's a good idea to make a note to where you can read more. For example PyTrends[1] or maybe even a reference[1], for the end of your report. This comes back to the general lack of structure.

Having tried, and failed, to run your code (spend about an hour working on it), I strongly recommend you give it a thorough rework as well, as I was unable to verify your methods. When you have two functions, which are doing exactly the same thing, but for different vaccines, they should be merged into a single, general method. Your code, as supplied in the archive file, cannot run by itself. There's dependencies on files which are missing (*stop_words_modified.txt*, *all_data_end.xls*). Furthermore, having no comments and much of the code inactive was no help.

I mention this, as you are including a certain code fragment directly in the report. Which opts me to try and confirm your statements in this section.

[1] https://github.com/GeneralMills/pytrends

Putting data into one document

Probably the trickiest part of the project, was to make sure the weekly and monthly data could coexist, and your method of 4.5 weeks allows this to happen. However, looking at the code, I recommend not hard-coding values for weekly csv files, as they may change. Rather, you should, at runtime, find a way to determine if a file is either on a monthly or weekly basis, this can be easily achieved, for example by using regular expressions.

Method used for prediction

When mentioning useful functions, even if their names may give a clear indication of what they do, I suggest adding a small description to them. Remember to once again make a reference to sklearn.

I would like some more information on how you got your results. All you mention is using the default parameter, OLS method and KFold. What happens in the folds, or maybe an example of what a *predict*() call returns may be interesting, as a plot maybe?

Findings and Conclusion

Having these in a table would make it look nicer. It's a shame you only got the result for a single vaccine, especially with all the extra code you wrote. Giving the RMSE is okay, but I suggest you compare these to the ones in the handout. Are you worse or better than them? And what can you conclude from these values? I would like more Findings if possible, is RMSE enough? As is, I would like to know what you achieved in the project. What was the point? Remember, the reader is not necessarily a fellow student.

In recap

Use the latex template and tighten up both language and content.
Use citation, if your work is dependent on the work of others, they should get credit.
Give more examples (plots, different score values etc), or make the code executable (include missing files), it's impossible to evaluate your work based purely on the RMSE of a single vaccine.
Include the other vaccine as well.
Rework your code. Don't rely on you to do anything manually. Make your code fully automatic, from getting the csv files all the way to your have a RMSE value, it should be scalable and require minimum effort to adjust for other vaccines. Add comments!.

References

[1] Author, Website, etc

WS 2016 Project 2

Martin Simon Haugaard
cdl966@alumni.ku.dk

1. INTRODUCTION

The objective in the project was to analyze reviews on Lego products, and feedback on these reviews, mined from an online forum[1]. The goal being, being able to determine the sentiment of a forum post, entirely based on mined forum posts, which had manually been evaluated.

2. METHODOLOGY

This section briefly covers the main aspects of the project.

2.1 Data

In this project, the data of interest are topics and responses to topics on a European online forum, which is generally devoted to reviewing different Lego sets. As a general observation, the users of the forum can be expected to be generally of the same target group, as they are all users of the same international online forum. This observation is critical, as it allows for sentiment evaluations to assume a somewhat general method, using the same language, and somewhat same method of formulating, any internal phrases in regards to Lego sets are understood forum wide etc. Without such knowledge making sentiment evaluations would have been much harder, as different target groups can formulate themselves in drastically different ways.

2.2 Data Mining

Initially, a spreadsheet of 4.900 forum posts were mined and shared via Google Drive with the participants of a Masters Course in Web Science at the University of Copenhagen. Each student then took part in helping to classify the posts, giving them a sentiment value of either -1 (Negative), 0 (Neutral) or 1 (Positive).

In my approach I then copied the spreadsheet, to safeguard from any further edit, and build a Python API using gspread[2] to automatically access the data. Once in my local domain, I strove to keep my computing offline, not depending on webservices to produce my data.

Of the sentiment mined, 3977 proved useful, as some were not evaluated, and some were evaluated differently than -1,0,1. 2218 (55.77%) sentiments were valued positively, 363 (9.1%) negatively and 1396 (35.1%) were neutral. This is worrisome, as any model which needs to be taught how to observe negative sentiment, having less than 10% of the total training data be negative, may not be enough.

2.3 Sentiment Evaluation

Three different methods of preforming sentiment evaluations were examined in this project.

2.3.1 SentiStrength

Initially a great amount of effort was spend on the free to use tool SentiStrength[3], as it offers both sentiment analysis on a sentence based level, and includes a suite of tools for further analysis. SentiStrength also takes some care in processing data which may include misspellings, emoticons and other elements which are a natural part of an online forum. Sadly, though, I found SentiStrength lacking, both in speed and accessibility, and furthermore SentiStrength depends on it's sentiment to be twofold, each sentence having both a positive and negative evaluation, which makes great sense, but sadly was intuitively possible with the given data.

2.3.2 uClassify

The second choice of approach were the tool uClassify[1], which comes in both online and offline versions, me choosing the latter. After having set up a socket server, and modifying code to accompany local access, three classes were made, negative, positive and neutral, and a 3-fold cross validation were preformed, in order to both train and teach uClassify to evaluate the forum posts, the results being elaborated in Section 3.

The method used by uClassify is fairly straight forward, as it will evaluate each sentence parsed, taking note of the provided sentiment and the words occurring in the sentence. Once a sufficient amount of training has been done, it can start making educated guesses on unseen sentences, weighing their value on classifying them as either negative, positive or neutral, based on their likeness on previous posts.

2.4 Deep Learning

Tools like SentiStrength and uClassify word on a naive basis, looking at words in a sentence and evaluating their

[1] www.eurobricks.com
[2] https://github.com/burnash/gspread
[3] http://sentistrength.wlv.ac.uk/

weight. However, this ignores much of the meaning in a sentence, such as grammar, phrases and so on, and while tools like SentiStrength has some manually coded options to account for this, such as a negating word in front of a positive *"Not Great"*, or slang *"h8- "hate"*, it cannot learn new information unless manually implemented.

Deep Learning aims to solve this problem, by looking at the entire sentence, and the structure within, in order to evaluate a sentence and determine its sentiment. Stanford University has produced a toolkit[2] for evaluating movie reviews. The tool is based upon Deep Learning, and allows for training, much like uClassify, however, in order to train it, trees has to be constructed from each sentence, as the structure of each sentence is now being evaluated. For example the positive sentence:

They look very nice

needs to turn into a tree, where each word has a weight, but the connection between words likewise hold weight

(3 (2 They) (3 (2 look) (3 (2 very) (3 nice))))

The numerical values are as follows:

0	1	2	3	4
Very Negative	Negative	Neutral	Positive	Very Positive

Once every review is on tree form, training can be preformed, and the new model can be tested. When training, it's an option to use Stanford's provided sentiment model to fill weigh each sentence, and only have the input provided tweak these values, or you can choose to construct an entirely independent model, based purely on the provided date. I experimented with both.

3. FINDINGS

First of. The initial uClassify method was able to correctly evaluate a given review roughly 60% of the time. Table 1 shows the number of correct (hit) and the wrong (miss) evaluations for each of the loops, correct as in matching with the Google Spreadsheet. These evaluations are of every sort,

Table 1: uClassify 3-fold cross validation results

Fold	hit	miss	success-rate
1	764	536	58.9%
2	780	520	60.0%
3	783	517	60.2%

positive, negative and neutral, and overall it's not a bad place to start. For a naive model this is what can be expected.

Secondly the Deep-Learning method was tested:
Without using the Stanford Sentiment data already in the toolkit to adjust the trees being constructed, I ended up only having a very few negative reviews when testing, as evident from the first fold's matrix produced when preforming 3-Fold testing, see Table 2. In the first fold, while there's no reviews expected to be 0 (Very Negative) or 4 (Very Positive), the training data did expect 870 Negative reviews, but ended up having 0. On the other hand there's far more Positive (7716) reviews compared to expected 5134. Overall I could say this method failed, and the only reason I get a fairly high hit rate [61.9%, 56.9%, 61.2%] respectively for each fold is due to the above average number of positive reviews in the data.

Table 2: Deep Learning - Without Movie Review Sentiment

Guess/Gold	0	1	2	3	4	Marg. (Guess)
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	52	176	241	0	469
3	0	818	2005	4893	0	7716
4	0	0	0	0	0	0
Marg. (Gold)	0	870	2181	5134	0	

Next, allowing the trees being generated to be affected by the sentiment already present in the Stanford Toolkit I get a much more varied set of results, as evident from another matrix from the 3-Fold testing, see Table 3.

Table 3: Deep Leaarning - With Movie Review Sentiment

Guess/Gold	0	1	2	3	4	Marg. (Guess)
0	0	0	0	0	0	0
1	114	2878	696	541	17	4246
2	0	225	1372	263	2	1862
3	7	424	254	1290	87	2062
4	0	1	0	4	10	15
Marg. (Gold)	121	3528	2322	2098	116	

This time when guessing (testing the training), both Negative, Neutral, Positive and Very Positive reviews are being presented, showing the effect of the movie review sentiment. For all three folds, their values and matrix see the Appendix.

4. CONCLUSION

As a standalone, the 4900 reviews, with invalid reviews excluded (bad/missing sentiment format), we were not given a large enough dataset to train without including pretrained data, such as the Movie Sentiment from the Stanford Toolkit. However, once included in the building of the data, the reviews give very credible feedback. A hit rate of 67%, without it being due to chance (as in the non-sentiment affected data) is a fair prediction rate, even if the Stanford boasts a 85% success-rate when analyzing movie reviews, our data is simply too small. The data being mined directly from a forum, where people often ends up misspelling, using emoticons and so on, only makes our limited dataset even weaker, as a persons unique way of formulating a review will cause outliers, which are hard to predict and learn from in isolation.

Overall, having a 7% better success-rate when using Deep-Learning is as expected, however I expect expanding the data set will improve success-rate even further.

5. REFERENCES

[1] uClassify classifier tool. https://www.uclassify.com/. Accessed: 2016-03-29.

[2] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

APPENDIX

Fold 1

```
EVALUATION SUMMARY
Tested 299379 labels
  261763 correct
  37616 incorrect
  0.874353 accuracy
Tested 8185 roots
  5550 correct
  2635 incorrect
  0.678070 accuracy
```

	0	1	2	3	4	
0	0	0	0	0	0	0
1	114	2878	696	541	17	4246
2	0	225	1372	263	2	1862
3	7	424	254	1290	87	2062
4	0	1	0	4	10	15
	121	3528	2322	2098	116	

Fold 2

```
EVALUATION SUMMARY
Tested 298018 labels
  262299 correct
  35719 incorrect
  0.880145 accuracy
Tested 8184 roots
  5549 correct
  2635 incorrect
  0.678030 accuracy
```

	0	1	2	3	4	
0	0	0	0	0	0	0
1	129	2761	586	716	34	4226
2	0	359	1482	279	4	2124
3	0	260	116	1294	127	1797
4	1	5	3	16	12	37
	130	3385	2187	2305	177	

Fold 3

```
EVALUATION SUMMARY
Tested 274860 labels
  241540 correct
  33320 incorrect
  0.878775 accuracy
Tested 8184 roots
  5472 correct
  2712 incorrect
  0.668622 accuracy
```

	0	1	2	3	4	
0	0	0	0	0	0	0
1	103	2116	319	382	14	2934
2	1	620	1802	291	4	2718
3	11	544	278	1552	145	2530
4	0	0	0	0	2	2
	115	3280	2399	2225	165	

Webscience Project 2 - Review
Assessment of:
449e3d6f886deec705c7842d2f5bdbc.zip

April 2, 2016

1. **Does the report answer correctly the project questions?**
   Yes, both a naive solution as well as a adeep learning solution have been tested using 3-fold cross-validation.

2. **Does the report break the project guidelines?**
   No, the report fully lived up to the guidelines.

3. **Are there portions of the report unrelated to the project questions?**
   The report is very concise and at no point diverges from the project questions.

4. **Are there sufficient examples to support the author's points?**
   All results are listed and referenced clearly, no points are made without sufficient data.

5. **Is the overall organization of the report clear and effective?**
   Yes, tables are clearly defined and referred, all sections contain only information relevant to the section.

6. **What are the report's main strengths?**
   It is very clear and concise, references are spot on, nice with data in appendix and references to this, a great quick overview of the different methods used. Readability is very high.

7. **What are the report's main weaknesses?**
   While it is noted that there is overwhelmingly many positive labels in our data, how many percent of these is not explicitly written. Writing that it is approximately 55% positive labels, diminishes the result of 60% and 67% hit rate, as they functionally only exceed by very little as opposed to just answering positive each time.

8. **Recommendations concerning the revision of this project**
   A description of the data in the beginning would be nice, this would help to show the significance of the results. Sections should never be empty as with **2. METHODOLOGY** and **2.2 Sentiment Evaluation**, a short intro here would be great. Remove "TEMPLATE"/"Keep anonymous" from title. Appendix is reffered to as Appendix A, there is no Appendix A.

9. **Summarization of**

   (a) **Technical quality**
   The code seems very well documented and very well organized, the imports are a bit of a mess though. All file names are hard coded, this is bad practise as it makes the code less reusable. I like that reformatted data is saved by pickling. Impressive that it automatically downloads data, great stuff.

   (b) **Presentation quality**
   The report presents the problem very well, as well as describes the way to attack the problem. Small mishaps as mentioned above

with the title, appendix and empty sections. But these are in my opinion very minor.

(c) **Adequacy of citation**
Everything that needed citation is cited, great job.

Amendment List

How many percent is positive labels in our data
This is now covered in section **2.2 Data Mining**

55% positive diminishes the result of 60% and 67%.
I would argue not, as I'm concerned about success-rates. Only correct predictions count, meaning any incorrect prediction should lower my result, still having 67% is quite good.

Description of the data in the beginning would be nice
A new section **2.1 Data** were introduced.

Sections should never be empty
Section **2 Methodology** how has brief introduction.

Short intro for 2.2 Sentiment Evaluation
Short intro added.

Remove "TEMPLATE"/"Keep anonymous" from title"
Removed, and now holds name of author.

Appendix A is missing (Just Appendix)
Fixed, now refers to appendix, no A.

Peer Review - Project Two

Anonymous

April 4, 2016

Peer-assessment

First I'd give some general feedback, and then I will quickly give feedback on each of your sections.

General Remarks

First of all, it's a nicely structured report, the sections are short and to the point, and everything seems to be covered.

You tend to use both I and We in your report, I find that highly distracting, I recommend you pick one or the other. There's some misspellings and grammatical errors in your report, for example, in **1. INTRODUCTION** the sentence *"In order label the sentiment"* is grammatically incorrect. You also vary between past and present tense (use / used), I recommend you do additional proofreading for the final hand-in.

I do not recommend including specific code into the main part of your report, how a tool functions may change with a new release, and it's generally not relevant for the average reader, even if the reader is a fellow student. Making a reference to the tools you're using will allow the reader to research specifics if need be. If you still need to show code, include them as an appendix, or make the scripts evident in the .tar.gz, which accompany the report.

Introduction

It introduces the reader to the problem you're trying to solve nicely. However, I dislike your use of references in this section, as it should be sufficient to have a reference after mentioning the dataset for the first time, and not after each of your examples, having this many references to the same source in this short a period is a messy. You have a reference to SentiStrength, but your reference is simply the word "SentiStrength", adding a hyper-link, author and/or paper is strongly recommended, I think you forgot this.

METHODOLOGY

SentiStrength

A quick introduction to SentiStrength could be nice, what is it? Why was is chosen versus other tools? Strengths and Weaknesses of SentiStrength?

I like your choice of weighing negative sentiment stronger, as it's rarer. You mention Maite Taboada here, include a reference so a curious reader and research further.

You talk about having reviews without annotations, I cold do with a short introduction to the dataset, who generated it? How large is it? Is this ignored part a relevant part? Remember the reader can be anyone with a scientific background, not necessarily just a fellow student or teacher.

Likewise, why do you need to re-annotate the data? Was it not sufficiently annotated?

In the introduction you said you would use 3-fold, and now you're using 10-fold. Why? SentiStrength supports 3-fold as well as 10-fold. I'm not satisfied with the reason given (better results? how so? Is our dataset large enough to give a sufficiently large sample size when 10-folding? What do the teachers ask for?)

You mention term weight at the very end of the section. I would have liked a better introduction to the method at work, with terms having weights. Who supplies the original weight? Can new words be introduced? It's not impossible to think that the forum from which our data is mined uses certain words to carry some weight, how can we teach these to SentiStrength?

Stanford

Just to be clear, Stanford also has 0 and 5 for sentiment values.

Skipping reviews which are too long is of course a valid option, however, as far as I can tell from your dataset file, you are now down to 3300 of the original 4900 reviews, just be careful you don't get a too small dataset. Also, reviews which are longer than 150 characters may be very relevant, as they are very likely the actual reviews, while texts which are smaller often are feedback to the actual reviews. Maybe another solution can be determined? For example. splitting long reviews, as the overall sentiment shouldn't change.

I like how you explain the general idea with a 3-fold approach, so much so that I recommend you move it to the start of the report, and maybe make a section (terminology) for terms like 3-fold etc which may not be given for every reader.

Findings

You value some of your results as "good"/"pretty good". What is good? Compared to what? (Stanford claims 85.4

If you don't make a reference to your tables, some readers may ignore them. Guide the reader, by making references to the correct table when needed, also Table 3 is

positioned as if it's part of the conclusion, it should appear in the Findings section.

You claim an average value of 72% in your SentiStrengh folds, I suggest you make a table holding these values, as you've done with Stanford CoreNLP.

Table 2: "Accuracy on the whole LEGO dataset using the training models from above". From above is a very weak reference, are you referring to text, or the table? Use \ref \label methods in latex to show the connection.

61% being not satisfying may need a bit of elaboration. How much is enough for it to be satisfying?

Conclusion

You mention making improvements, that's very general, which improvements? Which sections of your project would benefit from being improved?

What would be a satisfying result?

References

In reference [1] you spell lego with lower case, everywhere else LEGO is in capital font. [2] SentiStrength is hardly a reference.

In Recap

Overall a good first draft for a final report. I didn't get lost at any point, and I'm convinced the author has a good understanding of the methods described. I'm however missing some deeper evaluations and thoughts on the results, "pretty good" is not enough, I would love to see some elaborations on why you are/not satisfied with your results?

I did not execute your code while reviewing your assignment, as I assume it works, however, while skimming the code, I would've liked some comments and the amount of disabled code doesn't make the code any easier to understand. Is the disabled code irrelevant?