Task 3 – NLP Tasks

# NLP Task 1 – Name entity recognition

## Literature review

Name entity recognition has been used in a variety of papers for the purposes of brand name recognition. Within the Electronic commerce research journal an article was written that investigated name entity recognition of cross-border e-commerce titles based on TWs-LSTM. The TFIDF algorithm is used to manipulate the text corpus of the commodity to extract a weighted matrix, meanwhile the word2vec model represented the semantic meaning of the words extracted from the corpus’s bag of words. Where combined into a one-dimensional matrix that was passed into LTSM for commodity name entity recognition. The final accuracy achieved by this model was approximately 65% which was much higher than other models at the time [1]

## Rational

The proposed solution to the overall problem is to create an outlook system based on brands. Each review that was extracted utilising the webscraper had the model of the laptop along with the brand that it is made by. Name Entity recognition can be used to go through the review title for each review and extract this brand based on its classification. To accomplish this different entity name recognition systems can be utilised, or a model can be created and trained specifically for the desired purpose.

Both the spacey and Stanford NER models were used and compared, with the results from the one with the highest accuracy being appended to the dataset. Each of these have facilities to import custom labelled datasets and train the models based of them, which as will not be utilised as a labelled training set is not available, instead the pretrained models will be used.

The Stanford model utilises the conditional random field (CRF) algorithm which is a combination of the Hidden markov model and the Maximum entropy markov model. This algorithm assumes that features are dependent on each other which would be the case with a sentence as the context is created by the dependency of words onto each other [2]. Conditional random field algorithm makes use of this contextual information from the previous labels to better the current prediction [3].

Spacey utilises an unpublished algorithm but it utilises a word embedding method in conjunction with a deep convolutional neural network. This provides a model that has high adaptability and well-rounded accuracy. [2]

## Pre-processing

The review titles where able to be directly inputted into the models with no changes requires as they were only singular sentence inputs.

## Preliminary Analysis

The performance of the model will be based on the number of detections made and the correctness of those detections. Spacey has a higher detection rate compared to the Stanford model, however most of the detections from spacey are wrong. It recognises certain words as companies because of the use of capitalisation such as the detection of “Laptop Review” as an ORG when it is clearly not. When the results of the Stanford NER model are assessed, it has a lower detection rate but all the detections that are performed are correct.

Figure 1 – Brand name entity recognition results

Text

Description automatically generated

It should be noted that the spacey model is significantly faster than the Stanford model with it taking only 7 seconds to process the text as opposed to Stanford’s 22 minutes.

Comparing these results to the article assessed, it has the same level of performance with less complexity. The article is dated back to 2012 which in the period since, more sophisticated models have been developed with a much higher success rate.

## Code

CODE GOES HERE

# NLP Task 2 – Sentiment analysis

## Literature review

Sentiment analysis is a natural language processing method that detects if a body of text expresses a positive, negative or neutral outlook by scoring the words within the text and performing a summation. An article was included in the applied soft computing journal in 2020 that analysed the sentiment of tourism reviews, proposing the technique of sentiment padding. Sentiment padding creates a more consistent sample size and improves the proportion of sentiment information within each review. The article goes onto use neural networks; more specifically deep learning sentiment analysis models named lexicon integrated two-channel CNN-LTSM family models. This model combines CNN and LTSM/BiLTSM branches in parallel in order to achieve more accurate results. Utilising this approach on a variety of complex datasets, it was found to outperform many baseline methods. The final accuracy achieved was 50.68% on a reversed SST dataset while 95% accuracy was obtained on a Chinese review dataset. This was 2% higher than a zero-padding method. [4]

## Rational

The sentiment of a body of text can be directly linked to the outlook of the laptop. A positive sentiment can be assessed as a positive outlook for the laptop which is inline with the proposed solution. For this purpose, the Vader model will be utilised which is a lexicon and rule-based sentiment analysis tool that is attuned to sentiments expressed in social media [5]. As these reviews are based on opinions from a range of writers it should be well suited as social media sentiments are opinion based as well. Each section can be assigned a separate sentiment score which can then be assessed at a later stage based on what aspect of the laptop that the user is most interested in.

## Pre-processing

For this data as it was a large corpus some pre-processing was required. The first was the removal of any stop words from the text which added no additional meaning to the sentence such as joining words, this was done by tokenizing the words and removing any that appeared in NLTKs stop words list. Each segment of the data set besides the title was passed through this process (As the title was not used in this NLP task). Afterwards the non-text characters where removed such punctuation and brackets.

Lastly the corpus for each segment was normalized using lemmatization which reduce words to their base form. This acts to reduce errors when analysing sentiment as there are less variations of each word allowing for a better sentiment score to be assigned at the end, the NLTK WordNetLemmatizer was used to perform this.

## Preliminary Analysis

The vader package was used to apply a sentiment analysis on the processed corpus for each segment which was appended to the dataset. This final dataset with all the sentiment for each of the section was saved into a csv document. To assess the performance of this process a few sections of text will be selected and their corresponding sentiment scores. It will then be manually reviewed to gauge if the mapped sentiment matches the actual overall outlook of the body of text. For this the verdict paragraph will be selected as this gives a finalised opinion of the laptop which will be the easiest to assess the outlook off.

# Reference

[1] <https://link-springer-com.elibrary.jcu.edu.au/article/10.1007/s10660-019-09371-6>

[2] <https://blog.vsoftconsulting.com/blog/understanding-named-entity-recognition-pre-trained-models#:~:text=Model%20Architecture,while%20learning%20a%20new%20pattern.&text=In%20terms%20of%20performance%2C%20it,methods%20for%20entity%20recognition%20problems>.

[3] <https://medium.com/ml2vec/overview-of-conditional-random-fields-68a2a20fa541>

[4] <https://www-sciencedirect-com.elibrary.jcu.edu.au/science/article/pii/S1568494620303756?via%3Dihub>

[6] <https://analyticsindiamag.com/sentiment-analysis-made-easy-using-vader/>