Master MVA, 2017/2018

Algorithms for Speech and Natural Language Processing - TD 3

Sébastien Ohleyer

The goal of this assignment is to develop a normalisation system that can change raw English tweets into (partially) normalised tweets, suitable for further NLP processing. The system use two types of information for performing this normalisation task: contextual information and formal similarity information.

1 System

We made several choices to design this system. The first task was to clean the tweets, then we tokenized them and finally we perform the normalisation.

Group A quick look the corpus show that each line does not exactly correspond to one tweet. Sometimes tweets are on multiple lines. In order to deal with that, we decided to group every line which does not start with "RT". In other words, we considered that each tweet is a retweet. Ideally, the multiple tweet lines are grouped and form the orginal tweet and in the worst case we group two tweets. In both case, it allows us to create more context for each tweet and hopefully lead to better performances.

Clean Cleaning the tweets requires several operations. We decide to remove URLs, "RT", tags, line breaks, hex characters, hashtags and asterisks. For these operations, we use the `re` package. Then we convert every uppercase letters to lowercase letters to manage to cross with the dictionary. Finally, we tokenize every tweet using `nltk` package to remove punctuation marks.

Correct Now that tweets has been preprocessed, we can perform correction. For this purpose, we use contextual information using `context2vec` package and formal similarity with the Levenshtein distance (note that the edit distance had been coded by ourselves using dynamic programming).
For each word (token) in each tweet, we check whether it is in the lexicon provided by `context2vec` or not. If it is, we assume that the word does not need a correction. If not, we correct it by a naive approach consisting in two steps:

1. Compute the N best `context2vec` proposals, N can be seen as hyperparameter. If N is high, the correction will be slow but more efficient. Note that we used pre-trained context2vec model learned from UkWac. We develop a function, widely based on `eval_context2vec.py` to make it callable from our system.
2. We compute the Levenshtein distance between the incorrect word and the context2vec proposals and keep the closest word for correction.

2 Critics

After building the system, we test it on some tweets in the corpus. Note that depending on the value of N, the correction can be very long. Hence we proposed to perform correction only on a slice of the corpus. Please refer the README.md for more details on how to use the system and these different variables.

Errors Regarding to the output results, we observe that our system perform pretty badly. For example, some words are considered incorrect because they are proper noum like "Obama" or "Hollande". Hence, they are corrected even if they must not.

We can see that some really simple example such that "txi" are properly corrected in "taxi" but not everytime (we found an example where it was corrected in "x"). French words like "vendredi" were also badly handled as the lexicon only contains english words.

The final error we highlight concerns the context. After some trials of `context2vec` on some tweets of the corpus, we observe that the pre-trained model has some difficulties to detect the correct context and make some good proposals.

Improvements Obviously, some improvement can be made on our system. The first one concerns the model. It was trained on a completely different corpus and it is not really surprising that it does not perform well on ours. Tweets are a really particular framework and mix of language disturbs the system. Unfortunately, we could not do it because of computer performance limitation (needs for a GPU).

The second main improvement can be made on the correction. Indeed, we could use a ponderation between the proposal probabilites given by `context2vec` and the Levenshtein distances computed on these proposals. It could give a less naive approach for correction and increase performances. We did not follow this track by lack of time.