Bachelor of Science

Humboldt Universität zu Berlin

Wintersemester 2020/21

Semesterprojekt

Opinion Analyzer

Additional Proposal

- RNN for Topic Detection -

January 2021

1 Problem

Topic detection is a unsolved problem. We implemented a rudimentary baseline version based on word frequency, excluding the most frequent German words. The results are sobering. Also, stemmer and lemmarizer for German are mostly inadequate. There is a range of other options:

1. TF-IDF
2. Select word vectors with the highest distance to the mean of the text or a text body. As a modification, the mean might be calculated including word frequencies.

These are our fall back options.

2 Idea

We already have a very good dataset from Spiegel-Online with news articles and topics. This data is predestined to apply ML-techniques. Transformer are state of the art NLP tools, however, they rely on a maximum number of tokens. There are methods to circumvent this (e.g. Transformer-XL), however this complicated things in ways NLP beginner are not well suited for. Therefore, we would like to use a LSTM (RNN). This way we can use text of arbitrary lengths (ignoring vanishing gradient problems). There are a fixed number of word vector outputs, possibly including a confidence value. To build a loss function a concept from object detection and classification is applied. For each output the most similar (cosine similarity) and unique word vector to the topic tags is calculated. For all vectors with a corresponding tag, $\sum(1 - cosine)$ is used as a loss function. For samples where the output number exceeds the number of available topic tags, the distance to a NULL vector multiplied with the cosine distance to the closest (already used) topic tag is used. This way words similar to the topic tags are encouraged but duplicates are discouraged. At the same time NULL vectors, while possible, are not enforced since vectors with large distances to existing tags can yield small loss values too. This way additional topics exceeding the tags can be accepted. Pre-factors and other corrections, increasing or decreasing the importance of these values, will have to be introduced for fine tuning.

3 Requirements

Our experience with NLP and NNs is limited and the proposed idea, while intriguing, exceeds the scope of the *Semsterprojekt*. Therefore we ask for some help in finding suitable frameworks and tutorials to set up the above described model. Can it be implemented in flair? Are there other plug & play solutions to implement this? Are there models with a similar structure, that can be used either as a template or for transfer learning or even just the hyper parameters?