Natural Language Processing for tasks with specialized domain language – Capstone Proposal

data.camp097@audi.de

September 12, 2019

1 Domain Background

Natural Language Processing as a sub-discipline of machine learning has had major success in recent years on a broad range of natural language related tasks. An overview of recent trends is this field can be found in the paper ?Recent Trends in Deep Learning Based Natural Language Processing? by Young et al. However, when confronted with highly specialized sub-domain languages, commercial off the shelf (COTS) products decline in performance, often up to a point of being useless. As an example from personal experience, I would like to mention the use of Google Translate, a benchmark NLP product, on translating text from climbing guidebooks. The results are funny, but utterly useless. Another example more relevant for this work is the author?s area of expertise, automotive engineering. There exist many potentially useful NLP tasks in the automotive industry, ranging from automated scanners for the monitoring of (social) media with respect to potential quality or safety problems (a legal requirement for manufacturers), automated analysis of warranty and repair documents, systems for customer support, monitoring and analysis of patent applications to identify trends and many more. As in the example of climbing guidebooks, the results of COTS products when confronted with this form of specialized sub-domain language are usually poor. The topic of the capstone project is to make the first steps to remedy this situation. In particular, we will try to improve the word embeddings for a subdomain of interest.

2 Problem Statement

The basic hypothesis for this capstone project is the assumption that the dominant problem of NLP algorithms operating in highly specialized domain languages is the fact that during training the algorithm hasn?t seen any or enough domain specific text. One important element of most modern NLP pipelines is a dense vector representation, also referred to as word embedding, instead of sparse representations like one-hot-encoding or bag-of-words. Examples of these methods are word2vec, GloVe or FastText. I assume that in particular these representations are not trained well enough on domain-specific language.

3 Datasets and Inputs

As input I propose several sources of specialized technical literature: ? technical / specialist books ? Journal/conference papers ? patents and patent applications ? other specialist web resources Scraping and preprocessing this data will be part of the project as it will have to be automated. The scraping includes accessing publicly available databases for patents and online journals as well as the authors private collection of specialist literature (about 100+ books). The patent databases will be filtered by the classes of the CPC-classification scheme to only contain documents related to a particular field. The collected data will have to be preprocessed. Preprocessing includes cleaning the documents from things like stopwords, interpunctuation and capitalization. The preprocessed text will then be used for training of the dense vector representation.

4 Solution Statement

In a word: Train word embeddings on text from the subdomain of interest. First I propose to review the different existing word embeddings like Word2vec, GloVe or FastText and pick one to work with. An important part of the research will be the question of how to make sure the embedding performs well on both text in general *and* the domain language. To achive this, I aim at using some sort of pretraining. How this is done best to balance the performance is probably one of the key research questions. The implementation and training itself should be pretty straightforward as there are packages for many word embeddings are available that I would use for this project.

5 Benchmark Model

As a benchmark, I propose taking an existing trained word embedding. For all the mentioned word embeddings (Word2vec, GloVe, FastText) there exist extensively pretrained models for many languages (e.g. https://github.com/Hironsan/awesome-embedding-models). For details on how to get from a word embedding to an evaluation metric, please refer to section 6 Evaluation Metrics.

6 Evaluation Metrics

For vector representations, there are in general two different approaches for evaluation: extrinsic and intrinsic. Intrinsic evaluation takes into consideration only the embedding itself whereas extrinsic methods measure the embedding performance indirectly by a downstream NLP task with its own metric. In this thesis, I will use a classification task as

and measures for example the clustering and distance of synonyms and hyponyms, uses visual inspection of the embedding with t-SNE or similar projections.

Extrinsic evaluation measures the performance of NLP tasks using the embedding. The performance on the task is then taken as a measure for the quality of the embedding.

7 Project Design

The project design seems pretty straightforward. The steps themselves and the whose process will most likely be highly iterative. In a first step I will collect text for a particular specialized subdomain by collecting existing documents and scraping additional text off the internet. Where possible, I will try to leverage existing Python packages like pypatent etc. for the scraping In a second step, I will preprocess the collected texts. For this task, I will rely on existing NLP toolkits like NLTK. In a third step I will research existing word embedding / dense vector representations and pick one or two that seem to be state of the art and well suited for a range of tasks. I will in particular research the possibilities of pretraining or transfer learning to end up with a representation that does well on both text in general and the domain language in particular. In a forth step, I will train the vector representation of choice with the collected and preprocessed data and the training method chosen. In a fifth step, I will research in detail methods for evaluating word embeddings, both intrinsic and extrinsic. I will pick a few that seem to be suitable for this project in terms of effort, quality and value. In a sixth step, I will employ these metrics on the trained word embedding and compare it to a benchmark model.

Literature Tom Young, D. H. (n.d.). Recent Trends in Deep Learning Based. Retrieved from arXiv:1708.02709v8