Final Report
Team 31 "Practitioners"
Comparative Study of Neural Network based and Statistical models for Multi Domain Term Extraction

Varun Chhangani Vaishali Singh Nayanika

Introduction

Terminology extraction, also known as term extraction, is a subtask of information extraction. The goal of terminology extraction is to extract relevant words or phrases from a given corpus automatically. Our work will focus on the unsupervised automated domain term extraction method that considers chunking, pre-processing, and ranking domain-specific terms

Dataset description

In order to compare to a baseline, we train and test on the ACTER dataset utilised in the recent TermEval 2020 challenge. The domains wind energy and corruption represent the training set, dressage (equitation) the validation set, and heart failure the hold-out test set, for which the count of words and unique gold standard terms including named entities for English**.**
In the ACTER dataset, words were labelled as specific, common, and out-of-domain (OOD) terms, and named entities (NE). Specific terms are understood by domain experts, while common terms might also be additionally understood by laypersons. OOD terms might be specific to a different domain, but used in the domain at hand, e.g. statistical terms in the medical domain

ACTER

	Train	Validation	Test
Words	97145	51470	45788
Terms	2708	1575	2585

Baseline model

Term extraction using TF-IDF

Our baseline method is an unsupervised method and is based on TF-IDF. The basic underlying idea is that domain-specific terms occur in a particular domain with markedly higher frequency than they do in other domains, similar to term frequency patterns captured by TF-IDF. Hence, we compute TF-IDF from TF_{ij} , the term frequency of term i from documents in domain j, and IDFi , the inverse domain frequency. The calculation of TF_{ij} is via:

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$$

where n_{ij} is the number of occurrences of term i in the documents associated with domain j. IDF_i is calculated via:

$$IDF_i = log(\frac{|D|}{|\{d : t_i \in d\}|})$$

where t_i is the term, and D is the set of all domains. The final TF-IDF_{ij} value of a given term is the simple product of TF_{ij} and IDF_i .

We select those terms that have greater values than a particular criterion once the work of scoring phrases has been accomplished. We choose the threshold heuristically based on the score distribution, specifically the point when the TF-IDF scores drop significantly. That is, we take the current similarity as our threshold when the number of domain-specific terms gained at that similarity is less than 20% of the previously gathered domain-specific terms.

The threshold is found by using cross-validation technique over F1-score.

Token Classifier

The architecture we use for experimentation classifies each token of an input sentence separately, utilising the same fully connected layer for all tokens after they have been processed by RoBERTa. Because each sentence is only processed once by RoBERTa, this results in a large reduction in training and inference time. This architecture is commonly used in tasks like Named Entity Recognition (NER), which requires classification of each word in a sequence.

The model's input now comprises just of the sentences from the document that we want to process. The model then assigns one of three possible output labels to each input token: "B-T" for the start of a term, "T" for the continuation of a term, or "n" if the token is not part of a term.

We employed a classifier head as a linear layer with 768 input channels and 3 labels output. This was added to a pre-trained RoBERTa model. 12 layers each having:

A self attention layer, and three linear layers:

- One Attention output layer of input and output feature size 768
- One Intermediate wherein input features size 768, and output features are of size 3072
- One Layer output which re-combines the 3072 features to 768.

Hyperparameters:

- Optimizer - Adam
- Learning rate - 2*10^-5
- Batch size - 4 per GPU (trained on 2 GPU)
- Epochs - 5

- Loss Function - Cross Entropy

The loss function used is Cross Entropy, which follows:

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i$$

The Optimizer used is Adam which combines second order loss along with the first order for faster convergence. It also uses momentum to prevent getting stuck in a local minima

$$For\ each\ Parameter\ w^j$$
$$(j\ subscript\ dropped\ for\ clarity)$$

$$\nu_t = \beta_1 * \nu_{t-1} - (1 - \beta_1) * g_t$$
$$s_t = \beta_2 * s_{t-1} - (1 - \beta_2) * g_t^2$$

$$\Delta\omega_t = -\eta \frac{\nu_t}{\sqrt{s_t + \epsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

$\eta : Initial\ Learning\ rate$
$g_t : Gradient\ at\ time\ t\ along\ \omega^j$
$\nu_t : Exponential\ Average\ of\ gradients\ along\ \omega_j$
$s_t : Exponential\ Average\ of\ squares\ of\ gradients\ along\ \omega_j$
$\beta_1, \beta_2 : Hyperparameters$

Results

Comparative study

F1 Scores	**TF-IDF**	**Token Classifier**
Same Domain	**0.13**	**0.68**
Cross Domain	**0.05**	**0.53**

A significant result is the substantial improvement of F1- score of the token classifier over the baseline model TF-IDF. The score is better in the same domain as compared to the cross domain in both the models.

TF-IDF Output

Same domain:

The planning manager (Kretinga county municipality manager) Valerijones erneckis, senior architect of Kretinga county municipality department of Architecture and Urbanistics Jonas Petrulis, representative of detailed plan organizers Rolandas Rumšas, and planner Aušra Debolskyte participated at the meeting.

Extracted Terms: ['erneckis', 'Valerijones', ')', '(', 'municipality', 'county', 'Kretinga', 'planning', 'The', 'manager']

Cross domain:

it shall constitute a forum for the exchange throughout the EU of information on effective measures and experience in the prevention and combating of corruption;

Extracted Terms: ['for', 'forum', 'a', 'constitute', 'shall', 'it']

Token Classifier Output

Same domain:

What is Belgium doing against corruption?
['n', 'n', 'B-T', 'n', 'n', 'B-T', 'n']
Extracted Terms: {'Belgium', 'corruption'}

Cross domain:

Between the periods 1996-1998 and 2005-2007, the prevalence of HF after AMI declined from 28.1 % to 16.5%, with an adjusted odds ratio of 0.50 (95% CI, 0.44 to 0.55).
['n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'B-T', 'n', 'B-T', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'B-T', 'T', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n', 'n']
Extracted Terms: {'AMI', 'HF', 'odds ratio'}