Coleridge Initiative - Show US the Data

Strong, Careful, Simple: Solution Report

Mikhail Arkhipov

June 2021

Introduction

I would like to thank the organizers and people involved in data labeling for this exciting competition. I think the problem posed is extremely important in the context of the exponential growth of the data field.

A1. Brief Summary

- Private Leaderboard Score: 0.558
- Public Leaderboard Score: 0.623
- Third place on the Private Leaderboard
- Name: Mikhail Arkhipov
- Location: Moscow, Russia
- Email: arkhipovmu@gmail.com

A2. Background

A2.1 My Personal Background

Unrelated to the competition:

- BS and MS degree in Digital Signal Processing (DSP)
- Rich experience in development of DSP algorithms

Related to the competition:

- Development of open-source Natural Language Processing library DeepPavlov [1]

[1] DeepPavlov: `https://github.com/deepmipt/DeepPavlov`

- First Russian BERT model [2]
- Research in the field of Named Entity Recognition[3][4]

A2.2 Reasons to take part in the competition

Firstly, I think the problem is remarkably important. Extracting datasets from publications can be used for detection of new datasets, estimation of the impact of existing ones, and tracking progress in approaches for specific data.

Secondly, my NLP background seems to be relevant to the competition. However, most of my experience is related to Deep Learning, and I see a gap in my skills of building simple rule/regex-based solutions. The setting of the competition seemed to be a nice opportunity to develop such skills.

A2.3 Time Spent on the Development

I estimate the time spent on this competition as about four hours a day for about six weeks.

A3. Summary

The solution is based on pattern matching. Both for train and test documents, all capitalized sequences followed by brackets and containing data-specific keywords are extracted from the text to form a candidates set. All candidates containing stopwords (mainly specifying organizations) are discarded. Furthermore, the candidates that co-occur with word **data** less than in 10% of the cases are also discarded. During the evaluation, the collection of candidates is cleaned from brackets and used for simple string matching on test documents. The whole process takes several minutes.

A4/A5. Feature Selection / Engineering. Training Methods

The most important feature used in the solution is capitalization of words. Sequences of capitalized words (the first letter of the word is capital) cover the majority of dataset mentions. I also allow capitalized words to connect through prepositions or conjunctions. While this simple heuristic provides high recall, it suffers from low precision due to the abundance of technical terms and organization names.

[2] Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language `https://arxiv.org/abs/1905.07213` (100+ citations)

[3] Tuning Multilingual Transformers for Named Entity Recognition on Slavic Languages `https://www.aclweb.org/anthology/W19-3712.pdf`

[4] Application of a Hybrid Bi-LSTM-CRF model tothe task of Russian Named Entity Recognition `https://arxiv.org/pdf/1709.09686.pdf`

To improve precision, the candidates are filtered by leaving only those which include on of the following keywords:

- Study
- Survey
- Assessment
- Initiative
- Data
- Dataset
- Database

This requirement might seem too strict, however, it provides a fairly high precision-recall trade-off. Moreover, the list can be easily extended.

Further improvements in precision can be obtained by the exclusion of candidates containing stopwords. Stopwords mainly refer to organizations or tools. The following listing contains all stopwords used in the solution.

```
lab, centre, center, consortium, office, agency, administration,
clearinghouse, corps, organization, organisation, association,
university, department, institute, foundation, service, bureau,
company, test, tool, board, scale, framework, committee, system,
group, rating, manual, division, supplement,  variables, format,
documentation
```

The candidate is rejected if its lowercase version has any of the stopwords as a substring. The lab stopword is used with preceding white space to prevent matching inside other words.

Moreover, I restrict sequences to end with parenthesis that enclose abbreviations. It partially prevents merging multiple dataset mentions connected via conjunctions.

Finally, I drop parenthesis from retrieved mentions and perform a search over the entire available data (both train and test). I count all matches for each mention along with how often it occurs with the "data" word. Mentions that frequently co-occur with the word "data" more likely to be a dataset. The relation between total mention count and its count when the data word occurs in the same sentence is used to drop non-dataset substrings. I leave only those mentions that present in texts with the word "data" at least in 10 percent of the cases. I also drop mentions that present in texts less than two times.

The final prediction is a simple substring search for all extracted and filtered mentions without brackets for each document. In addition, for each detected mention I look for parenthesis with abbreviations right after the mention. Sometimes they contain specified versions of the dataset (e.g. NELS:88). The abbreviations from the parenthesis are added as a separate predictions.

Both for pattern matching (extraction of dataset mentions) and substring search (getting predictions) a simple sentence tokenizer that splits by dot symbol is used. I also refused to use regular expressions because while working fine on train set they sometimes hangs on the private part of the test (I lost a number of submissions due to this issue).

A6. Interesting findings

The first thing I find interesting and want to highlight is the relation of the problem to existing NLP problems. Detection of dataset mentions is quite similar to Entity Linking problem, specifically, to the entity (mention) detection stage [1].

Furthermore, there is some evidence [2] that in a low-resource regime simple methods like ”collect a dictionary of all entity mentions strings from Wikipedia and use it for simple string matching” works surprisingly well for Entity Linking. Even application of modern sophisticated neural methods shows significantly lower results in absence of such ”mention strings collection”, while providing noticeable improvement when both neural and string matching approaches are used.

I believe that it is also possible to adopt methods similar to classic Hearst Patterns [3] to solve the task. Hearst Patterns are used to extract hypernyms from the raw texts. The following list contains some examples of these patterns:

- X which is a (example, class, kind, ...) of Y
- X (and, or) (any, some) other YX which is called Y
- X a special case of Y

This patterns shows impressively good results compared to modern neural methods [4]. These findings motivated me to try similar patterns for the Colerige competition. Examples of the patterns I experimented with:

- (data, samples, collected, obtained) from Xxx Xxx (XXX)
- XXXX dataset ... datasets: Xxx Xxx, Xxx Xxx, and Xxx

However, I was not able to achieve high scores using these patterns. Nevertheless, a simple ”data from Xxx Xxx” pattern provides really good coverage (estimated by manually inspecting the results), but, I hypothesize, not high enough precision.

Two findings of the success of simple methods described above motivated me to focus on the simple string/pattern matching heuristics. I believe that with some additional efforts devoted to extending patterns and improve cleaning my method can become a very strong baseline. Moreover, I think that focus on pattern matching without spending a lot of time on gathering external resources and implementation of neural approaches was a key to get the high final leaderboard position.

I would also like to highlight that provided markup complicates implementation of some straightforward approaches like tagging with neural network. A possible alternative to provided setting would be exhaustive markup of a limited number of documents. In such case it would be much easier to apply modern techniques.

A7. Simple Features and Methods

Being extremely simple itself the solution can be further sped up by adding index retrieval to check only relevant documents in the evaluation stage. In this case, only sentences that contain all words from the candidate dataset mention would be processed by the substring search. This should reduce inference time by an order of magnitude.

References

[1] Ozge Sevgili et al. "Neural entity linking: A survey of models based on deep learning". In: *arXiv preprint arXiv:2006.00575* (2020).

[2] Shyam Upadhyay, Nitish Gupta, and Dan Roth. "Joint multilingual supervision for cross-lingual entity linking". In: *arXiv preprint arXiv:1809.07657* (2018).

[3] Marti A Hearst. "Automatic acquisition of hyponyms from large text corpora". In: *Coling 1992 volume 2: The 15th international conference on computational linguistics*. 1992.

[4] Stephen Roller, Douwe Kiela, and Maximilian Nickel. "Hearst patterns revisited: Automatic hypernym detection from large text corpora". In: *arXiv preprint arXiv:1806.03191* (2018).