Entity relationship ranking using differential keyword-role affinity

Rohit Naini
Capital One
McLean, VA
rohit.naini@capitalone.com

Pawan Yadav
Capital One
McLean, VA
pawan.yadav@capitalone.com

ABSTRACT

Identifying relationship between named entities from a corpus of text is a well studied NLP problem. In this paper, we consider a tractable version of this wherein sample text snippets and corresponding roles are extracted and need to be ranked on relevance of text to the role. Our scoring approach uses a cumulative estimated relevance of all keywords observed in the text snippet. Relevance metrics are computed based on differential affinity of keywords to the roles observed in the training data.

KEYWORDS

Entity role ranking, keyword affinity

ACM Reference format:
Rohit Naini and Pawan Yadav. 2017. Entity relationship ranking using differential keyword-role affinity. In *Proceedings of DSMM'17, Chicago, IL, USA, May 14, 2017,* 2 pages.
https://doi.org/http://dx.doi.org/10.1145/3077240.3077255

1 INTRODUCTION

Building machine models to understand natural language in spoken and text form is gaining importance with commercialization of voice assistants like Google Home, Alexa, Cortana etc. Identifying roles and relationships between financial entities using text filings is an adaptation of this larger "machine understanding and concept modeling" problem. The scoring task itself makes the broader problem more tractable by extracting entities and roles. The significant challenge in this task is using the highly unnatural and legalistic wording observed in these filings to build a good language model.

Our approach for this task abandons complicated language and concept modeling tasks. We instead dissolve the sentence structure and focus on identifying keywords within the extracted text and perform relevance estimation of these keywords for roles specified in the training data. For the scoring part, we build a Maximum-likelihood estimate for the given role to observe keywords seen in the extracted "three sentences". The score itself is the Naive-Bayes probabilistic estimate of said role being associated with all the keywords.

DSMM'17, May 14, 2017, Chicago, IL, USA

ACM ISBN 978-1-4503-5031-0/17/05…$15.00
https://doi.org/http://dx.doi.org/10.1145/3077240.3077255

The remainder of this paper delves into the specifics of Training methodology, scoring technique used and concludes with a brief summary on our results and potential risks and suggested improvements for future iterations.

2 TRAINING METHODOLOGY

In order to build an estimate of keyword role affinity, we use Python to perform data ingestion and processing. Our choice is guided by the simplicity and richness of the toolsets that Python provides through Pandas and NLTK framework. Our training process consists pre-processing, keyword extraction and affinity estimation as detailed here:

2.1 Pre-processing

Our pre-processing component involves the following steps

- Ingest training data files from all entities
- Concatenate all data into a single dataframe
- Identify some rows (<10) for which no expert rating is provided and discard from training data
- Normalize the role description from plural, capitalized versions into standardized role descriptors for each of the 10 classes
- Convert text relevance ratings into numeric $\{2, 1, 0, -1\}$ and average in case of multiple expert label ratings

2.2 Keyword extraction

Once the ratings are estimated for training, our next step involves processing the column "three sentences" to extract keywords for further processing. Steps involved are as follows

- Strip all control characters and convert text to lowercase
- Split text into a word array per row and combine into a single array
- Use NLTK and limit keywords based on the english dictionary, and remove common stopwords

We observe a drastic reduction in keyword count after controlling for uniqueness and stopwords. Some reduction is also observed due to non-dictionary words like bullet labeling and abbreviations.

2.3 Role affinity estimation

For each of the observed keywords in the training data, we need to build and estimate of the propensity to be associated with the labeled roles. In order to this, we limit our association to average expert rating to strictly relevant text snippets, ignoring neutral and irrelevant text extracted.

Corresponding to each keyword in the training data, we build a vector of observed frequencies with each keyword. This observed

frequency serves as an empirical conditional probability estimate after normalization. One additional caveat for this process is that we add an $\epsilon = 0.01$ probability as a heuristic simplification to the unobserved roles to avoid saturation to 0 or 1 of cumulative probability during scoring.

For observed keywords w_k, for $k \in \{1, 2, 3, ..., K\}$ and roles r_i, for $1 \leq i \leq 10$. We have a $K \times 10$ affinity probability matrix P, where p_{ki} is the empirical conditional probability of observing keyword w_k in association with role r_i

$$\sum_{i=1}^{10} p_{ki} = 1, \quad \forall 1 \leq k \leq K$$

For scoring purposes, the probability matrix P serves as the training artifact that captures the entity role affinity.

3 TEST DATA SCORING

The scoring task involves ranking the identified entities and roles in test data in decreasing order of relevance. The metric Normalized Discounted Cumulative Gain (NDCG) used to rate scoring task rely exclusively on the ordering within each original entity, role group. Our probabilistic estimates provide a natural setup to do this ranking.

Within each text snippet "three sentences" in the test data, we use the cumulative probability over all observed keywords(also seen during training) while ignoring the rest. We use the Bayesian probability of observing these keywords for the labeled role class.

To formalize this scoring process. Suppose for a test data row, we observe N keywords w_{j_i} where $j_i \leq K, \quad \forall 1 \leq i \leq N$. The Bayesian probability estimate q_s corresponding to specified role r_s is given by

$$q_s = \frac{\prod_i^N p_{j_i s}}{\sum_s \prod_i^N p_{j_i s}}$$

Once we compute the Bayesian probability estimates for all rows within a filing entity role class, we simply sort in decreasing order of these probabilities.

4 RESULTS AND SUMMARY

On scoring the test data, we observe that the resulting probability estimates are highly polarized due to the cumulation process. Important thing to note however is that since the evaluation metrics care purely about the row ordering and are agnostic to the scoring distribution, our approach still works. We identify this polarization is due to the size of the training data, we fail to observe some keywords in more than one role. The heuristic imputation of 0.01 probability is an attempt to regularize the scoring process and prevent it from saturation. Other potential scoring techniques and use of weights based on TF-IDF metrics can help improve performance by elevating the novel and meaningful keywords.

Another limitation of our technique is that observed scorable keywords vary in number across multiple rows. We might be able to improve performance by using a keyword count based normalization per row.

Our submission is a first attempt using a probabilistic estimation framework towards solving a highly complex natural language problem. Future directions for this work include making the keyword extraction process more robust using parts of speech tagging and usage of TF-IDF metrics for test data scoring.

REFERENCES

[1] Louiqa Raschid, Doug Burdick, Mark Flood, John Grant, Joe Langsam, Ian Soboroff, and Elena Zotkina. Financial Entity Identification and Information Integration (FEIII) Challenge 2017: The Report of the Organizing Committee. In *Proceedings of the Workshop on Data Science for Macro-Modeling (DSMM@SIGMOD).*